R+D 5854-EN-01

2

AD

AD-A213 278

FLOOD INUNDATION MODELLING USING MILHY

Interim
Final Technical Report

by

L.Baird and M.G.Anderson

Volume 2

June 1989

European Research Office
U.S. Corps of Engineers
London England

CONTRACT NUMBER DAJA 45-87-C-0053

Dr.M.G.Anderson

CONTENTS

VOLUME II

APPENDIX I

MILHY3 : PROGRAMMING DETAILS

Laura Baird
University of Bristol

May 1989

Synopsis: Details of MILHY computer code. The subroutines are documented in the same order in which they appear in the programme. Functions are documented after the subroutine.

Contents:

PROGRAMME CHANGES SINCE MILHY2

I have made major changes in MILHY2 (coded by Dr S Howes, University of Bristol) to improve the predictive capability of the downstream conveyance estimation, specifically under out-of-bank conditions. , Therefore, turbulent exchanges of momentum between segments of flow in a cross-section and multiple-routing reaches have been incorporated.

However, the overall structure of the MILHY3 code remains unchanged since MILHY2. All new computations occur within existing subroutines, some of which have been extensively rewritten. The subroutines which have significantly changed are:-

CMPRC
ROUTE
PRTHYD

Minor changes have occurred in subroutines:-

CMPHYD
ADHYD
STHYD
STT
BLKDTA

To allow for the nnew computations, BLKDTA has been amended to allow more data to be read from 'data1', and a new common block, 'COMMON/BLOCK2' has been added to allow this new data to be transferred between subroutines.

The definition of some arrays in 'COMMON/BLOCK1' have been changed. These are:-

A(20, 70)	end area
Q(20, 70)	discharge
DEEP (20, 70)	elevation of water surface
C(20, 6)	absolute elevation

The array size of A, Q and DEEP has been increased to allow for the generation of rating curves for each segment of flow across a section, necessary if multiple routing reaches are invoked. The identification number of a segment is firstly the cross-section identification number, followed by the segment number (numbered from left to right across the section looking downstream).

The size and definition of the array C (20, 6) has been changed. It was previously carried in 'COMMON/BLOCK 1' as the travel time coefficient, but is now the water surface elevation plus minimum cross-sectional elevation. The travel time coefficient is now carried in COMMON/BLOCK2 as CC(20).

ROIN has to be redefined to ensure that the runoff volume is correctly interpreted. In MILHY2 the array ROIN was the hydrograph volume divided by the subcatchment area, hence giving the depth of inundation over the catchment. With the introduction of multiple routing reaches, however, it was found that there was some difficulty in carrying the correct catchment area due to the repeated ROUTE command. For simplicity, therefore, catchment area was not included in the computation of ROIN and in MILHY3 ROIN is the total volume of discharge (i.e. the area under the hydrograph).

Before application of MILHY3 the user is recommended to study the dataset details for data 1, as there are significant changes in the manner in which the commands need to be entered and changes in the order of the data entries.

The programme code and dataset have been written to enable the user to choose which of the conveyance methods are most suitable for the application. Guidelines for the selection of these methods are given in the final report, DAJA ... ...

SUBROUTINE NAME: MAIN

SYNOPSIS: Opens files for input and output. Initialises certain variables. Calls appropriate subroutine according to command.

COMMAND:

INPUT: Two data files called 'data1' and 'data2' must exist.

'data1' - contains programme controls and data
'data2' - additional information for the infiltration algorithm

See appendix I and II for details of these two data files

OUTPUT: Opens the output file 'results' to which the details of the simulation are to be sent.

VARIABLES USED:

Variables held in common block 'BLOCK1':

OCFS(300,6)	discharge
DATA(311)	data input for each command
CFS(300)	unit hydrograph discharge
CTBLE(50,11)	command table
RAIN(300)	cumulative precipitation at equal time intervals
ROIN(6)	volume of discharge hydrograph
A(20,70)	end area (for rating curve)
Q(20,70)	discharge (for rating curve)
DEEP(20,70)	elevation of water surface (for rating curve)
ITBLE(50,2)	integer table
DP(20)	flow depth for previously computed travel time flow relationship
SCFS(20)	discharge for previously computed travel time flow relationship
C(20,6)	absolute stage elevations (for rating curve)
ZALPHA(20)	alphanumeric code table
IEND(6)	number of points in the hydrograph
DA(6)	drainage area
DIST(6)	segment boundary point for each segment of a floodplain and chennel cross-section
SEGN(6)	Mannings 'n' for segment of a floodplain and channel cross-section
DT(6)	time increment for precipitation or discharge data
PEAK(6)	peak discharge of hydrograph
ISG(6)	last elevation input for each segment
NPU	punch code
NHD	identification number of hydrograph
NER	error

MAXNO	maximum number of data entries for any one command
NCOMM	number of command
ICC	continuation card
NCODE	command number
TIME	simulation start time
KCODE	measurement unit of input data if (KCODE=0) then imperial units else metric
ICODE	required measurement unit of output if (ICODE=0) then imperial units else metric

Variables held in common block 'BLOCK2':

PERQ(20,70)	percentage discharge for segment of a floodplain and channel cross-section (for rating curve)
TQ(20,6)	total discharge for cross-section (for rating curve)
CC(20)	travel time coefficient for previously computed travel time relationship
LL(6)	number of zero discharge values for segment of a floodplain and channel cross-section
INRC	identification number for upstream segment rating curve
LRC	identification number for downstream segment rating curve

CONSTRAINTS: The most important constraints in this program involve the limits to array size which are dimensioned in COMMON. For example, program can only hold 6 hydrographs or 6 rating curves at any one time. 17 commands which are defined in BLOCK DATA are used by MILHY3 (as in the original form of HYMO). The legal MILHY values for these are provided in HONDO and appendix I.

CALLED BY:

SUBROUTINES CALLED:
HONDO
STHYD
RECHD
CMPHYD
PRTHYD
PUHYD
HPLOT
ADHYD
SRC
CMPRC
STT
CMPTT
ROUTE
RESVO
ERROR
SEDT

FUNCTIONS CALLED:

NOTES There are changes to the array size of some variables and changes to the definition of variables in COMMON/BLOCK1 from MILHY2.

SUBROUTINE NAME: HONDO

SYNOPSIS: Reads in command and associated data from file 'datal' and by comparison to the legal commands contained in CTBLE (initialized in BLOCK DATA), it determines the command number (NCODE). It collects up the variables from the variable format data field.

COMMAND:

INPUT: Data is read in from file 'datal'.
Command must be located in the first 20 columns on each line, and is read in variable ALPHA (11) (FORMAT 2A1,9A2). The data must be in columns 21 to 80, and is read into variable CHAR (60) (FORMAT (60A1).

Legal commands are:

START
STORE HYD
RECALL HYD
COMPUTE HYD
PRINT HYD
PUNCH HYD
PLOT HYD
ADD HYD
STORE RATING CURVE
COMPUTE RATING CURVE
STORE TRAVEL TIME
COMPUTE TRAVEL TIME
ROUTE
ROUTE RESERVOIR
ERROR ANALYSIS
SEDIMENT YIELD
FINISH

Additional legal entries to file 'datal' include:

'*' in column 1 - if the line is a comment line
'*' in column 80 - if a new page is required for output

OUTPUT: Writes out the command and associated data to file 'results', and returns the value of NCODE to MAIN which is then used to select the next subroutine to be called. All data which has been collected for the command is held in common, in the array DATA (311).

VARIABLES USED: Variables held in common plus

CHAR(60)	data and associated text
ALPHA(11)	command
AUXA(10)	array used to collect up data
AUXB(10)	array used to collect up data

CONSTRAINTS:	The form of the data file 'datal' must be strictly adhered to. HONDO will not tolerate spelling mistakes. The command and data must also be entered into the correct columns. The data must be in the order which is expected by HONDO (these details are provided in appendix I).
CALLED BY:	MAIN
SUBROUTINES CALLED:	
FUNCTIONS CALLED:	GIT

NOTES

SUBROUTINE NAME;	STHYD
SYNOPSIS:	A model control procedure. Stores the coordinates of a measured hydrograph and adds a constant baseflow discharge to all data points.
COMMAND:	
INPUT:	The data input for this command has been read into DATA(311) by HONDO and is transferred from this array into the following variables which are used in this subroutine: ID NHD DT(ID) DA(ID) BSF OCFS(300,ID)
OUTPUT:	Stores discharge hydrograph (time and associated discharge values), runoff volume, and peak discharge: OCFS(300,ID) ROIN(ID) PEAK(ID)
VARIABLES USED:	Variables held in common plus

ID	storage location number
NHD	hydrograph identification number
BSF	baseflow discharge
DUMMY(300)	discharge values converted to metric units

CONSTRAINTS:	Only 6 hydrographs at any one time can be stored by this program. In any one hydrograph, a maximum of 300 points are allowed.
CALLED BY:	MAIN
SUBROUTINES CALLED:	
FUNCTIONS CALLED:	
NOTES:	If the addition of baseflow is not required, a zero value must be entered.

SUBROUTINE NAME: RECHD

SYNOPSIS: Model control procedure.
Recalls previously computed and punched hydrographs.

COMMAND: RECALL HYD

INPUT: The data input for this command has been read into DATA(311) by HONDO and is transferred from this array into the following variables which are used in this subroutine:

ID
NHD
DT(ID)
DA(ID)
PEAK(ID)
ROIN(ID)
IEND(ID)
OCFS(300,ID)

OUTPUT: Stores hydrograph OCFS(300,ID)

VARIABLES USED: Variables held in common plus

ID	storage location number
NHD	Hydrograph identification number

CONSTRAINTS: Only 6 hydrographs at any one time can be stored by this program. In any one hydrograph, a maximum of 300 points are allowed.

CALLED BY: MAIN

SUBROUTINES CALLED:

FUNCTIONS CALLED:

NOTES:

SUBROUTINE NAME: CMPHYD

SYNOPSIS: Hydrological procedure.
Developes unit hydrograph and convolves it with incremental runoff to produce the discharge hydrograph. Runoff is derived by calling subroutine SOILM, utilizing the infiltration, or using the curve number routine

COMMAND: COMPUTE HYD

INPUT: Data has been read into DATA(310) by HONDO and is transferred from this array into the following variables which are used in this subroutine:

ID
NHD
DT(ID)
DA(ID)
CN
HT
XL
RAIN(300)

OUTPUT: Stores the calculated discharge hydrograph, runoff volume, and peak discharge

OCFS(300,ID)
ROIN(ID)
PEAK(ID)

VARIABLES USED: Variables held in common plus

ID	storage location number
NHD	hydrograph identification number
HT	difference in elevation
XL	length of main channel

CONSTRAINTS: A maximum of 6 hydrographs can be stored.
A maximum of 200 data can be included in the precipitation data.

CALLED BY: MAIN

SUBROUTINES CALLED: SOILM

FUNCTIONS CALLED:

NOTES: This subroutine has been updated to permit the user to select the curve number routine or the infiltration algorithm.

SUBROUTINE NAME: SOILM

SYNOPSIS: Simulation of infiltration and hence incremental runoff associated with a particular storm event, and redistribution of soil water after precipitation ceases. Includes a stochastic methodology for incorporating spatial variability of soil hydrological properties.

COMMAND:

INPUT: Certain data has been passed from CMPHYD to SOILM:

DT(ID)
IR
CUMRAIN

Remaining variables are read directly into SOILM from data file 'data2'. Appendix II provides the details of the form of this data file and the information which is required by SOILM.

OUTPUT: Provides incremental runoff which is located in DATA(311) and which is passed back to CMPHYD. This runoff is at the same time interval as the precipitation data which has been supplied (DT(ID)).

VARIABLES USED:

DT(ID)	Time increment for precipitation and hence runoff data
IR	number of rainfall observations
CUMRAIN(251)	cumulative rainfall totals
TIME	time when simulation begins
SIMDUR	simulation duration (hours)
ALR	rain start time (hours)
AMR	rain stop time (hours)
AF	simulation iteration period (seconds)
NLA	number of cells in layer 1 in soil column
NLB	number of cells in layer 2 in soil column
NL	number of cells in soil column
TCOM(20)	thickness of each cell (metres)
NSCOL	number of soil columns
IPCAREA	percent are occupied by soil column
SR1	soil water content at saturation, layer 1 in soil column
SR2	soil water content at saturation, layer 2 in soil column
SR3	soil water content at saturation, layer 3 in soil column

ASR1 ASR2 ASR3	Same variable definitions as the three above, but variable types are DOUBLE PRECISION rather than REAL
SSR1	Standard deviation of SR1
SSR2	Standard deviation of SR2
SSR3	Standard deviation of SR3
SATCON	saturated hydraulic conductivity (met-es per second) layer 1
SATCON2	saturated hydraulic conductivity (metres per second) layer 2
SATCON3	saturated hydraulic conductivity (metres per second) layer 3
ASATCON ASATCON2 ASATCON3	Same variable definitions as the three above, but variable types are mean values in DOUBLE PRECISION rather than REAL
SSATCON1	Standard deviation of SATCON1
SSATCON2	Standard deviation of SATCON2
SSATCON3	Standard deviation of SATCON3
DETCAP	surface detention capacity (metres)
ADETCAP	DOUBLE PRECISION surface detention capacity
SDETCAP	Standard deviation of detention capacity
THETA(20)	initial soil water content for each cell (cubic metres per cubic metres)
ATHETA(20)	DOUBLE PRECISION initial soil water content (cubic metres per cubic metres) mean value
STHETA	Standard deviation of THETA(20)
NQ	number of observations on soil moisture characteristics curve
X(20)	moisture values on soil moisture characteristic curve for layer 1 (cubic metres per cubic metres)
Y(20)	suction values on soil moisture characteristic curve for layer 1 (metres)
X2(20)	moisture values on soil moisture characteriscic curve for layer 2 (cubic metres per cubic metres)
Y2(20)	suction values on soil moisture characteristic curve for layer 2 (metres)

	X3(20)	moisture values on soil moisture characteristic curve for layer 3 (cubic metres per cubic metres)
	Y3(20)	suction values on soil moisture characteristic curve for layer 3 (metres)
	AX(20) AX2(20) AX3(20)	Same variable definitions as the X(20), X2(20), and X3(20) above, but variable types are mean values in DOUBLE PRECISION rather than REAL
	SCURV1	Standard deviation of soil moisture characteristic curve for layer 1
	SCURV2	Standard deviation of soil moisture characteristic curve for layer 2
	SCURV3	Standard deviation of soil moistire characteristic curve for layer 3
	EMAX	maximum evaporation during the day (metres per second)
	WT	write-out time interval (hours)

	IOUT	determines amount of output
		if (IOUT=1) total output
		if (IOUT=0) shorter output

CONSTRAINTS: A maximum of 10 soil columns for any one subcatchment area is permitted.
The soil moisture characteristic curve can be defined by up to a maximum of 20 points.
The soil column can have a maximum of 20 cells.
The initial soil moisture contents, defined for each cell at the start of simulation, must lie within the range of the soil moisture characteristic curve.

CALLED BY: CMPHYD

SUBROUTINES CALLED:
HYDCON
TWO
GRAD
SMCURV
GO5DDF(NAG subroutine)

FUNCTIONS CALLED:
RMAX
RMIN

NOTES:

SUBROUTINE NAME:	HYDCON	
SYNOPSIS:	Calculates hydraulic conductivity for a particular layer in the soil column from the soil moisture characteristic curve, using the Millington and Quirk method.	
COMMAND:		
INPUT:	Variables passed from SOILM: X(20) Y(20) SATCON SR	
OUTPUT:	Unsaturated hydraulic conductivity values are passed back to SOILM in Z(20).	
VARIABLES USED:	X(20)	moisture values on soil moisture characteristic curve for the particular layer (cubic metres per cubic metres)
	Y(20)	suction values on soil moisture characteristic curve for the particulat layer (metres)
	SATCON	saturated hydraulic conductivity for the particular layer
	SR	saturated soil moisture content for the particular layer
	Z(20)	unsaturated hydraulic conductivity values corresponding to X(20) above.
CONSTRAINTS:	Maximum points on the soil moisture characteristic curve, and hence the hydraulic function is 20.	
CALLED BY:	SOILM	
SUBROUTINES CALLED:		
FUNCTIONS CALLED:		
NOTES:		

SUBROUTINE NAME:	TWO	
SYNOPSIS:	Calculates the soil water pressure, hydraulic potential, and hydraulic conductivity for each cell in the soil column, associated with a particular soil water content.	
COMMAND:		
INPUT:	Variables passed from SOILM: NA NB G(20) GZ(20) Z(20) X(20) Y(20) DEPTH(20)	
OUTPUT:	Soil water pressure, hydraulic potential, and hydraulic conductivity are passed back to SOILM SWP(20) HPOT(20) COND(20)	
VARIABLES USED:	NA	number of cells in layer 1
	NB	number of cells in layer 2
	THETA(20)	initial soil moisture content of each cell
	G(20)	gradient of soil moisture characteristic curve, ie grad between each pair of points
	GZ(20)	gradient of hydraulic function, ie grad between each pair of points
	X(20)	moisture values on soil moisture characteristic curve for the particular layer (cubic metres per cubic metres)
	Y(20)	suction values on soil moisture characteristic curve for the particular layer (metres) values
	Z(20)	unsaturated hydraulic conductivity values corresponding to X(20) above
	DEPTH(20)	distance from surface to the midpoint of each cell
	SWP(20)	soil water pressure of each cell
	HPOT(20)	hydraulic potential of each cell
	COND(20)	conductivity of each cell
CONSTRAINTS:	A maximum of 20 cells in the soil column is permitted	
CALLED BY:	SOILM	

SUBROUTINES
COMMAND:

FUNCTIONS
COMMAND:

NOTES:

SUBROUTINE NAME:	GRAD	
SYNOPSIS:	Calculates the gradient of the soil moisture characteristic curve, and hydraulic conductivity function.	
INPUT:	Variables passed from SOILM: X(20) Y(20) Z(20)	
OUTPUT:	Variables containing gradients passed back to SOILM. G(20) GZ(20)	
VARIABLES USED:	X(20)	moisture values on soil moisture characteristic curve for the particular layer (cubic metres per cubic metres)
	Y(20)	suction values on soil moisture characteristic curve for the particular layer (metres) values
	Z(20)	unsaturated hydraulic conductivity values corresponding to X(20) above
	G(20)	gradient of soil moisture characteristic curve, i.e. gradient between each pair of points
	GZ(20)	gradient of hydraulic function, i.e. gradient between each pair of points
CONSTRAINTS:	A maximum of 20 cells in the soil column is permitted	
CALLED BY:	SOILM	
SUBROUTINES COMMAND:		
FUNCTIONS COMMAND:		

NOTES:

SUBROUTINE NAME:	SMCURV	
SYNOPSIS:	Generates new soil moisture characteristic curve based on the randomly generated moisture values.	
INPUT:	Variables passed from SOILM: AX(20) Y(20) SCURV SR NQ	
OUTPUT:	Coordinates of new soil moisture characteristic curve passed back to SOILM: XNEW(20) YNEW(20)	
VARIABLES USED:	AX(20)	values of soil moisture on input soil moisture characteristic curve DOUBLE PRECISION variable type
	Y(20)	values of suction on input soil moisture characteristic curve
	SCURV	standard deviation of soil moisture characteristic curve in DOUBLE PRECISION
	SR	saturated soil moisture content
	NQ	number of coordinates defining soil moisture characteristic curve
	XNEW(20)	generated soil moisture content on new soil moisture characteristic curve
	YNEW(20)	generated suction values on new soil moisture characteristic curve
CONSTRAINTS:	A maximum of 20 points to define the soil moisture characteristic curve	
CALLED BY:	SOILM	
SUBROUTINES CALLED:	GO5DDF (NAG subroutine)	
FUNCTIONS CALLED:	RMIN RMAX	

<u>NOTES:</u>

SUBROUTINE NAME;	PRTHYD
SYNOPSIS:	Model control procedure. Prints out the coordinates of a hydrograph and/or the peak value and runoff volume. Converts OCFS(300,ID) to a stage array, S(300,ID) using a recalled rating curve.
COMMAND:	PRINT HYD.
INPUT:	The data input for this commandn has been read into OCFS(300,ID) by HONDO and is transferred from this array into the following variables which are used in this subroutine: ID NPK IDR IN Details of the hydrograph are held in common and are referenced by ID.
OUTPUT:	Discharge, DUMMY(300) or stage, S300,ID) hydrograph are written to output file 'results'. DUMMY(300) S(300,ID) ROIN1 PEAK1 PEAK5
VARIABLES USED:	Variables in common plus

ID	storage location number
NPK	form of output required 0 peak and volume only 1 discharge hydrograph 2 stage hydrograph
IDR	identification number of rating curve or segment to be used for conversion to a stage hydrograph
IN	format of output 0 five columns across page 1 single column
DUMMY(300)	discharge array (converted to metric units if required)
S(300,ID)	stage array (converted to metric units if required)
PEAK1	peak discharge
ROIN1	volume of hydrograph
PEAK5	peak stage

CONSTRAINTS:	Maximum of 300 points define the hydrograph. For conversion to stage hydrograph, rating curve must have been computed. A stage hydrograph cannot be computed if multiple routing is invoked.
CALLED BY:	MAIN
SUBROUTINES CALLED:	
FUNCTIONS CALLED:	

NOTES:

Conversion to a stage hydrograph uses a previously computed rating curve.

SUBROUTINE NAME:	PUHYD
SYNOPSIS:	Model control procedure. Punches hydrograph coordinates on cards so that they can be recalled by RECHD
COMMAND:	RECALL HYD
INPUT:	The data input for this command has been read into DATA(311) by HONDO and is transferred from this array into the following variables which are used in this subroutine: ID Details of the hydrograph are held in common variables and are referenced by ID.
OUTPUT:	Punched version of OCFS(300,ID)
VARIABLES USED:	Variables held in common plus ID — storage location number DUMMY(300) — discharge values converted to metric units
CONSTRAINTS:	Maximum of 300 points define the hydrograph. Maximum of 6 hydrographs in store at any one time.
CALLED BY:	MAIN
SUBROUTINES CALLED:	
FUNCTIONS CALLED:	

NOTES:

SUBROUTINE NAME:	HPLOT
SYNOPSIS:	Model control procedure. Plots either 1 or 2 hydrographs on a set of axis.
COMMAND:	PLOT HYD.
INPUT:	The data input for this command has been read into DATA(311) by HONDO and is transferred from this array into the following variables which are used in this subroutine:- ID1 ID2 Details of the 2 hydrographs are held in common variables and are references by ID1 and ID2
OUTPUT:	Discharge plots and axis are written to output file 'results'. CFS(300)
VARIABLES USED:	Variables in common plus ID1 ID2
CONSTRAINTS:	If the time interval of the two hydrographs to be plotted is not equal, the larger increment is selected and the other hydrograph points are interpolated at this increment.
CALLED BY:	MAIN
SUBROUTINES CALLED:	
FUNCTIONS CALLED:	

NOTES:

SUBROUTINE NAME:	ADHYD
SYNOPSIS:	Model control procedure Adds together the coordinates of two hydrographs
COMMAND:	ADD HYD
INPUT:	The data input for this command has been read into DATA(311) by HONDO and is transferred from this array into the following variables which are used in this subroutine: ID NHD ID1 ID2 Details of the 2 hydrographs are held in common variables and are referenced by ID1 and ID2
OUTPUT:	The discharge coordinates, peak discharge, and runoff volume of the resultant hydrograph: OCFS(300,ID) PEAK(ID) ROIN(ID)
VARIABLES USED:	Variables in common plus ID — storage location number for resultant hydrograph NHD — hydrograph identification number of resultant hydrograph ID1, ID2 — storage location numbers of the two hydrographs to be added
CONSTRAINTS:	If the time interval of the two hydrographs to be added is not equal, then the smaller increment is selected and the other hydrograph points are interpolated at this increment.
CALLED BY:	MAIN
SUBROUTINES CALLED:	
FUNCTIONS CALLED:	

NOTES:

SUBROUTINE NAME:	SRC
SYNOPSIS:	A model control procedure Stores a rating curve in form of elevation, end area, discharge table
COMMAND:	STORE RATING CURVE
INPUT:	The data input for this command has been read into DATA(311) by HONDO and is transferred from this array into the followjng variables which are used in this subroutine: ID VS DEEP(20,ID) A(20,ID) Q(20,ID)
OUTPUT:	Stores the rating curve in variables held in common: DEEP(20,ID) A(20,ID) Q(20,ID)
VARIABLES USED:	Variables held in common plus ID — storage location number of rating curve VS — valley cross section number
CONSTRAINTS:	Only 6 rating curves can be held within the program at any one time. Maximum number of points defining rating curve are 20.
CALLED BY:	MAIN
SUBROUTINES CALLED:	
FUNCTIONS CALLED:	

NOTES:

SUBROUTINE NAME: CMPRC

SYNOPSIS: A hydrological procedure.
Computes rating curve for valley cross section using Mannings equation. If turbulent exchange routines are invoked calculates rating curve incorporating momentum transfer between channel and floodplain flows during out-of-bank conditions. If multiple routing reaches are invoked calculates separate rating curves for each segment of the cross-section and computes the percentage of total flow which would occur in each segment at the twenty stage computation points.

COMMAND: COMPUTE RATING CURVE

INPUT: The data input for this command has been read into DATA(311) by HONDO and is transferred from this array into the following variables which are used in this subroutine:

ID
IT
MR
VS
NSEG
ELO
EMAX
SLOPE1
SLOPE2
SEGN(NSEG)
DIST(NSEG)
DATA(10:310)

OUTPUT: Stores the rating curve and percentage flow in each segment in variables held in common

A(20,ID)
Q(20,ID)
C(20,ID)
DEEP(20,ID)
PERQ(20,ID)
TQ(20,ID)

VARIABLES USED: Variables held in common plus

ID storage location number for rating curve
IT turbulent exchange between main channel
and floodplains invoked
MR multiply routing reaches invoked
VS valley section identification number
NSEG number of segments in valley section
ELO lowest elevation

EMAX	maximum elevation
SLOPE1	channel slope
SLOPE2	flood plain slope
DATA (10:310)	alternate distances and elevations (defining cross section)

CONSTRAINTS: Maximum number of segments in a cross section is 6. Maximum number of points in a rating curve is 20. Turbulent exchange and multiple routing reaches may be invoked independently but for either to operate there must be a floodplain segment on either side of channel segments. For accuracy the user is recommended to have cross-sectional positional data at both extremities of flow segments.

CALLED BY: MAIN

SUBROUTINES CALLED:

FUNCTIONS CALLED:

NOTES: The user is recommended to consult MILHY3: datal on the form of dataset 'datal' must take for application of multiple routing reaches.

SUBROUTINE NAME:	STT
SYNOPSIS:	Model control procedure. Stores a depth, flow, travel time table (used in flood routing).
COMMAND:	STORE TRAVEL TIME
INPUT:	The data input for this command has been read into DATA(311) by HONDO and is transferred from this array into the following variables which are used in this subroutine: ID REACH XL SLOPE MET1 DP(20) SCFS(20) CC(20)
OUTPUT:	Stores travel time table in following common variables: DP(20) SCFS(20) CC(20)
VARIABLES USED:	Variables held in common plus ID storage location number REACH reach identification number XL length of reach SLOPE slope of reach
CONSTRAINTS:	A maximum of 20 points are allowed to define a travel time table.
CALLED BY:	MAIN
SUBROUTINES CALLED:	
FUNCTIONS CALLED:	

NOTES:

SUBROUTINE NAME:	CMPTT
SYNOPSIS:	Hydrological procedure. Compute travel time table.
COMMAND:	COMPUTE TRAVEL TIME
INPUT:	Data has been read into DATA(3110) by HONDO and is transferred from this array into the following variables which are used in this subroutine: ID REACH NOVS XL SLOPE MR INRC LRC
OUTPUT:	Stores the travel time table in following common variables: DP(20) SCFS(20) CC(20)
VARIABLES USED:	Variables held in common plus

ID	storage location number
REACH	reach identification number
NOVS	number of valley sections in the reach
XL	length of reach
SLOPE	Slope of reach
MR	multiple routing invoked
INRC	upstream segment rating curve identification number
LRC	downstream segment rating curve identification number

CONSTRAINTS:	A maximum of 20 points are allowed to define a travel time table. A maximum of 6 valley sections are permitted in a reach, except where multiple routing reaches are invoked where two segment section must be identified.
CALLED BY:	MAIN
SUBROUTINES CALLED:	

FUNCTIONS CALLED:

NOTES: If multiple routing reaches are invoked, a compute travel time table and route command must be entered for each segment routing reach.

SUBROUTINE NAME: ROUTE

SYNOPSIS: A hydrological procedure.
Routes a hydrograph through a reach using the variable storage coefficient method.
If multiple routing reaches are invoked, routes a hydrograph through a segment routing reach.
Also calculates inflow hydrograph for segment reach using PERQ(20) from rating curve.

COMMAND: ROUTE

INPUT: The data input for this command has been read into DATA(311) by HONDO and is transferred from this array into the following variables which are used in this subroutine:

ID
NHD
IDH
DT(ID)
MR

Details of the hydrograph to be routed are held in common variables and are references by IDH.
Details of the inflow segments rating curve are held in common variables and are referenced by:

PERQ(20)
TQ(20)
C(20,INRC)
INRC.

OUTPUT: Stores the calculated outflow hydrograph, its peak discharge, and runoff volume in common variables:

OCFS(300,ID)
PEAK(ID)
ROIN(ID)

Proportional discharge for each time increment value (if multiple routing reaches are invoked) written to output file 'results'.

DOCF(300,ID)

VARIABLES USED: Variables held in common plus

ID	storage location number of calculated outflow hydrograph
NHD	hydrograph identification number of outflow hydrograph

IDH	storage location number of inflow
hydrograph	DT(ID) iteration period of outflow
hydrograph	
MR	multiple routing invoked
DOCFS	dummy discharge area to prevent overwriting of inflow array
P	percentage of inflow in multiple routing reach segment

CONSTRAINTS: Discharges included in the inflow hydrograph must be within the limits of the travel time table, otherwise there is no way to define the travel time coefficient. If the solution to the routing equations fails to converge after 10 iterations, convergence is forced. If multiple routing reaches are invoked the inflow hydrograph must not exceed the rating curve used to compute proportional inflow in segment.

CALLED BY: MAIN

SUBROUTINES CALLED:

FUNCTIONS CALLED:

NOTES: If multiple routing reaches are invoked, a compute travel time table and route command must be entered for each segment routing reach. Also the identification number of the inflow and outflow hydrographs must not be the same.

SUBROUTINE NAME:	RESVO
SYNOPSIS:	A hydrological procedure. Routes hydrograph through a reservoir.
COMMAND:	ROUTE RESERVOIR
INPUT:	The data input for this command has been read into DATA(311) by HONDO and is transferred from this array into the following variables which are used in this subroutine: ID NHD IDH SCFS(20) STORE Details of the inflow hydrograph are held in common variables and are referenced by ID: DT(ID) DA(ID)
OUTPUT:	The calculated outdlow hydrograph, peak discharge, and runoff volume is stored in common variables: OCFS(300,ID) PEAK(ID) ROIN(ID)
VARIABLES USED:	Variables held in common plus

ID	storage location number of calculated outflow hydrograph
NHD	hydrograph identification number of outflow hydrograph
IDH	storage location number of inflow hydrograph
SCFS(20)	discharge values of the storage discharge relationship defined for the reservoir
STORE	storage values of the storage discharge relationship defined from the reservoir

CONSTRAINTS:	The discharge of the inflow hydrograph must be within the storage discharge relationship defined from the reservoir. A maximum of 20 points are allowed to define this relationhip.
CALLED BY:	MAIN

SUBROUTINES
CALLED:

FUNCTIONS
CALLED:

NOTES:

SUBROUTINE NAME:	ERROR
SYNOPSIS:	Model control procedure. Calculates a number of indices or objective functions which detail the degree of fit between two hydrographs.
COMMAND:	ERROR
INPUT:	The data input for this command has been read into DATA(311) by HONDO and is transferred from this array into the following variables which are used in this subroutine: ID1 ID2 Details of the 2 hydrographs are held in common variables and are referenced by ID1 and ID2
OUTPUT:	The values of HYMO's original error statistics plus an additional 13 objective functions are written to output file 'results'. ESDEV PCTER OF1 OF2 OF3 OF4 OF5 OF6 OF7 OF8 OF9 OF10 OF11
VARIABLES USED:	Variables in common plus

ID1	storage location number of first hydrograph (usually assumed to be measured)
ID2	storage location number of second hydrograph (usually assumed to be calculated)
ERR	error (measured - calculated discharge)
ESDEV	error standard deviation
PCTER	percentage peak discharge error
OF1	absolute sum of errors
OF2	ordinary least squares
OF3	log of ordinary least squares
OF4	relative sum of errors

OF5	absolute error difference
OF6	relative error difference
OF7	absolute error divided by variance
OF8	relative error divided by variance
OF9	absolute error difference divided by variance
OF10	relative error difference divided by variance
OF11	Pearsons correlation coefficient

CONSTRAINTS: The first hydrograph (ID1) is taken to be the measured. All indices are printed out in file 'results' in metric units.

CALLED BY: MAIN

SUBROUTINES CALLED:

FUNCTIONS CALLED:

NOTES:

SUBROUTINE NAME: SEDT

SYNOPSIS: A hydrological procedure.
Computes sediment yield for a field using the Universal soil loss equation.

COMMAND: SEDIMENT YIELD

INPUT: The data input for this command has been read into DATA(311) by HONDO and is transferred from this array into the following variables which are used in this subroutine:

ID
SOIL
CROP
CP
SL

Details of the hydrograph are held in common variables and are referenced by ID:

ROIN(ID)
DA(ID)
PEAK(ID)

OUTPUT: Writes out the sediment yield to the output file 'results':

SED

VARIABLES USED: Variables held in common plus

ID	storage location number of hydrographs
SOIL	soil erodibility factor
CROP	the cropping management factor
CP	the erosion control practice factor
SL	the slope length and gradient factor

CONSTRAINTS:

CALLED BY: MAIN

SUBROUTINES CALLED:

FUNCTIONS CALLED:

NOTES:

FUNCTION NAME: GIT (TCARD, J, JLAST, SHIFT).

SYNOPSIS: Converts alphabetic array to floating point number.

INPUT: TCARD(10)
J
JLAST
SHIFT
A(10)

OUTPUT: GIT

VARIABLES USED: TCARD
J
JLAST
SHIFT
A(10)

CONSTRAINTS:

CALLED BY: HONDO

NOTES:

FUNCTION NAME: RMAX(X,NQ)

SYNOPSIS: Returns the maximum element in a REAL array.

INPUT: X(NQ) X is a REAL array of size NQ

OUTPUT: RMAX

VARIABLES USED: X(NQ)
RMAX

CONSTRAINTS:

CALLED BY: SOILM

NOTES:

FUNCTION NAME: RMIN(X,NQ)

SYNOPSIS: Returns the minimum element in a REAL array.

INPUT: X(NQ) X is a REAL array of size NQ

OUTPUT: RMAX

VARIABLES USED: X(NQ)
RMAX

CONSTRAINTS:

CALLED BY: SOILM

NOTES:

SUBROUTINE NAME: BLOCK DATA

SYNOPSIS: Initializes certain variables. These variables are used to determine the number of commands, the command, the command number, the maximum number of data entries which are associated with the command, and the data which is entered in the variable format data entry area.

INPUT:

OUTPUT: Initialized arrays:

ZALPHA(20)
CTBLE(50,11)
ITBLE(50,2)
NCOMM

VARIABLES USED:

ZALPHA(20) alphanumeric code table containing:

1234567890 *.,-(filled with blanks)

CTBLE(50,11) command table containing:

START (filled with blanks)
STORE HYD
RECALL HYD
COMPUTE HYD
PRINT HYD
PUNCH HYD
PLOT HYD
ADD HYD
STORE RATING CURVE
COMPUTE RATING CURVE
STORE TRAVEL TIME
COMPUTE TRAVEL TIME
ROUTE
ROUTE RESERVOIR
ERROR ANALYSIS
SEDIMENT YIELD
FINISH
(filled with blanks)

ITBLE(50,2) integer table containing:

1 4
2310
3310
4310
5 4
6 1
7 2
8 4

```
 9100
10311
11100
12  8
13  7
14 25
15  2
16  5
17  0
```

NCOMM number of commands contains:

7

CONSTRAINTS:

CALLED BY:

SUBROUTINES CALLED:

FUNCTIONS CALLED:

<u>NOTES:</u>

APPENDIX 2 data1

The commands permitted by MILHY3 are the same as those for the original MILHY. They perform two functions: model control and hydrological procedures. The legal commands are:

Model control procedures:

START
STORE HYD
RECALL HYD
STORE RATING CURVE
STORE TRAVEL TIME
ADD HYD
PRINT HYD
PLOT HYD
PUNCH HYD
ERROR ANALYSIS
FINISH

Hydrological procedures:

COMPUTE HYD
COMPUTE RATING CURVE
COMPUTE TRAVEL TIME
ROUTE
ROUTE RESERVOIR
SEDIMENT YIELD

An '*' in column 1 means that the line is a comment.
An '*' in column 80 means skip to a new page before writing to file.

The command must be in columns 1-20, and the data in 21-80.

The variables in the data area MUST be in the correct order, but can be surrounded by text (i.e. units and variable labels).

The following two pages list the variables which are required for each command, and the order in which they are required.

Data requirements of HYMO3
(data to be located in 'data 1')

HYDROLOGICAL PROCEDURE	Variable used in subroutine
COMPUTE HYD	
Storage location number for hydrograph	ID
Hydrograph identification number	NHD
Time increment for rainfall data (hours)	DT(ID)
Watershed area (sqmi/km^2)	DA(ID)
Curve number (enter zero if not invoked)	CN
Watershed height, maximum difference (ft/m)	HT
Main stream length (mi/km)	XL
Rainfall, cumulative totals (inches/mm)	RAIN(300)
COMPUTE RATING CURVE	
Storage location number for rating curve	ID
Turbulent exchange of momentum between segments (not invoked enter zero) (invoked, enter 1-4 depending on method)	IT
Multiple routing reaches (not invoked enter zero) (invoked enter one)	MR
Valley section location number	VS
Number of segments in channel (max. of 6)	NSEG
Minimum elevation (ft/m)	ELO
Maximum elevation (ft/m)	EMAX
Channel slope	SLOPE1
Floodplain slope	SLOPE2
Manning 'n' for each segment (negative value for channel segments)	SEGN(NSEG)
Segment boundary points (horizontal distance) (ft/m)	DIST(NSEG)
Cross-section co-ordinates (distance then elevation) (ft/m)	DATA(12,311)

/Compute Travel Time

COMPUTE TRAVEL TIME

Storage location of travel time table	ID
Reach identification number	REACH
Number of valley sections in reach	NOVS
Reach length (ft/m)	XL
Slope (average for flow segments)	SLOPE
Multiple routing reaches (not invoked do not enter) (invoked enter one)	MR
Inflow rating curve identification	INRC
Outflow rating curve identification (first digit is storage location of the rating curve, the second digit is the segment number)	LRC

N.B. If multiple routing reaches not invoked do not enter values for INRC and LRC

ROUTE

Storge location number of outflow hydrograph	ID
Hydrograph identification number of outflow hydrograph	NHD
Storage location number of inflow hydrograph	IDH
Time increment (hrs)	DT(ID)
Multiple routing reaches (not invoked, do not enter) (invoked enter one)	MR

ROUTE RESERVOIR

Storage location number of outflow hydrograph	ID
Hydrograph identification number of outflow hydrograph	NHD
Storage location number of inflow hydrograph	IDH
Reservoir outflow storage relation (max 20 points)	DT(ID)

SEDIMENT YIELD

Storage location of number of hydrograph	ID
Soil, crop, conservation and gradient factors	SOIL, CROP, CP, SL

/Model Control Procedures

MODEL CONTROL PROCEDURES

Start

Start time (hours)	TIME
Punch code	NPU
(not invoked enter zero)	
(invoked enter one)	
Data input	KCODE
imperial enter zero	
metric enter one	
Data output	ICODE
imperial enter zero	
metric enter one	

STORE HYD

Storage location number for hydrograph	ID
Hydrograph identification number	NHD
Time increment for discharge data (hrs)	DT(ID)
Watershed area (sq.mi/km^2)	DA(ID)
Baseflow (added to discharge) (cfs/m^3s^{-1})	BSF
Discharge (300 points max.) (cfs/m^3s^{-1})	OCFS(300,ID)

RECALL HYD

Storage location number for hydrograph	ID
Hydrograph identification number	NHD
Time increment for discharge data (hours)	DT(ID)
Watershed area (sq.mi/km^2)	DA(ID)
Peak discharge (cfs/m^3s^{-1})	PEAK(ID)
Runoff volume (cf/m^3)	ROIN(ID)
Number of points in hydrograph	IEND(ID)
Discharge (cfs/m^3s^{-1})	OCFS(300,ID)

STORAGE RATING CURVE

Storage location number for rating curve	ID
Valley section number	VS
Rating curve points	
elevation (ft/m)	DEEP(20,ID)
end area (ft^2/m^2)	A(20,ID)
flow rate (cfs/m^3s^{-1})	Q(20,ID)

/Storage Travel Time

STORAGE TRAVEL TIME

Storage location number for travel time table	ID
Reach identification number	NHD
Length of reach (ft/m)	XL
Slope either channel or flood plain or weighted average of the two	SLOPE
Depth (ft/m)	DP(ID)
Discharge (cfs/m^3s^{-1})	SCFS(20)
Storage coefficient	C(20)

ADD HYD

Storage location number for resultant hydrograph	ID
Hydrograph identification number of resultant	NHD
Storage location of two hydrographs to be added	ID1, ID2

PRINT HYD

Storage location number of hydrographs	ID
Specification of type of output	NPK

0 peak and volume only (cfs/m^3s^{-1})
1 discharge hydrograph (cfs/m^3s^{-1})
2 stage hydrograph (ft/m)

Rating curve identification for conversion of hydrograph	IDR

PLOT HYD

Storage location number of the 1 or 2 hydrographs to be plotted	ID1, ID2

PUNCH HYD

Storage location number of hydrograph	ID

ERROR ANALYSIS

Storage location numbers of 2 hydrographs to be compared	ID1, ID2

FINISH

No information required

Setting up datal: changes from MILHY2

Listed in the previous few pages are the legal commands and variables for dataset 'datal'. Whilst the commands remain unchanged the attention of the user is drawn to order of the variables and the introduction of new variables. These new variables are all model control variables allowing the user to select the method of incorporating turbulent exchange between cross-sectional flow segments and to introduce multiple-routing reaches.

If the multiple routing reaches or the turbulent exchange algorithms are invoked along a particular reach then any channel cross-sectional segment in that reach must have floodplain segments on either side. In addition, if multiple routing is invoked there may be only two cross-sections along the reach. After the rating curves have been computed for these two cross sections each segment of flow must be separately routed. This means there must be the same number of segments in both the upstream and downstream cross-sections. For each segment a separate 'TRAVEL TIME' and 'ROUTE' command are entered and the outflow hydrographs from segment are added to give the total discharge across the cross section. Care must be taken to ensure storage location number (ID's) of inflow hydrographs are not reused in the application of multiple-routing reaches. The variables INRC and LRC in the 'TRAVEL TIME' command identify the inflow and outflow segments for one reach of a multiple-routing reach. These identification numbers consist of the cross-section ID number followed by the segment number, numbered left to right looking downstream. In the ROUTE subroutine the inflow hydrograph is distributed between the segments of the upstream cross-section using the rating curve identified by INRC. It is important therefore that the upstream cross-section is positioned at or near the upstream extremity of the reach.

An example 'datal' dataset is given for a single reach where both multiple routing and turbulent exchange algorithms are invoked.

```
C ====================================================================
C
C           MILHY3 - a mathematical flood forecasting model for
C                    ungauged catchments
C
C ====================================================================
C
C Program:      MILHY3
C               (MILHY2) including two-stage channel modelling.
C               With improved out-of-bank flood modelling incorporating
C               TURBULENT EXCHANGE between in and out of bank flows and
C               MULTIPLE ROUTING REACHES -allowing separate pathways for
C               channel and floodplain flows.
C
C Coded by:     L.Baird
C               University of Bristol
C
C NOTES         Altered subroutines
C                       ADHYD
C                       STHYD
C                       CMPHYD
C                       CMPRC
C                       STT
C                       CMPTT
C                       ROUTE
C                       PRTHYD
C                       BLKDTA
C
C Notes        The structure of the code remains unaltered.
C              All additional computations occur within existing
C              subroutines.
C              HOWEVER, there are significant changes in the manner in
C              which the dataset DATA1 must be set out to facilitate
C              utilisation of the new capabilities.
C
C UNITS        All computations (except in the infiltration algorithm)
C              are carried out in imperial units, irrespective of KCODE
C              and ICODE.  COMMON/BLOCK1 and COMMON/BLOCK2 use imperial.
C ---------------------------------------------------------------------


      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20,6),
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NHD,NER,MAXNO,NCOMM,ICC,NCODE,TIME,KCODE,ICODE
      COMMON/BLOCK2/ PERQ(20,70),TQ(20,6),CC(20),LL(6),INRC,LRC

C Definition of variables in common 1

C OCFS     Hydrograph discharge
C DATA     Data associated with each command
C CFS      Unit hydrograph discharge
C CTBLE    Command table
C RAIN     Cumulative precipitation values
C ROIN     Volume of discharge hydrograph
C          note: this variable is no longer divided by area
```

```
C A       End area
C Q       Flow rate for rating curve
C DEEP    Elevation of water surface (for rating curve)
C ITBLE   Integer table
C DP      Flow depth for previously computed travel time flow relationship
C SCFS    Discharge for previously computed travel time flow relationship
C C       Absolute stage elevations computed in rating curve
C ZALFA   Alphnumeric code table
C IEND    Number of points in the hydrograph
C DA      Drainage area
C DIST    Segment boundary point for each segment of a cross section
C SEGN    Mannings 'n' for each segment of a cross section
C DT      Time increment for rainfall or discharge
C PEAK    Peak discharge for hydrograph
C ISG     Last elevation input in each segment  position
C NPU     Punch code
C NHD     Hydrograph identification number
C NER     Error number
C MAXNO   Maximum number of data entires to be expected for any command
C NCOMM   Number of commands
C ICC     Continuation line
C NCODE   Number of command
C TIME    Start time of simulation
C KCODE   Measurement unit of input
C                0 - imperial
C            not 0 - metric
C ICODE   Measurement unit of output
C                0 - imperial
C            not 0 - metric
C Variables common 2
C PERQ   Percentage discharge (rating curve computation)
C TQ     Total discharge (rating curve computation)
C CC     Travel time  coefficient
C LL     Number of zero discharge values in rating curve segment
C INRC   Inflow rating curve (multiple routing)
C LRC    Outflow rating curve (multiple routing)

      OPEN (1,STATUS='old',FILE='data1')
      OPEN(25,FILE='data2',STATUS='old')
      OPEN(6,STATUS='new',FILE='results')
      NCODE=0
      NPU=0
      ICC=0
1     NER=0
      CALL HONDO
      IF (NER) 2,2,19
2     GO TO (3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19), NCODE
3     TIME=DATA(1)
      NPU=DATA(2)
      KCODE=DATA(3)
      ICODE=DATA(4)
      GO TO 1
4     CALL STHYD
      GO TO 1
5     CALL RECHD
      GO TO 1
6     CALL CMPHYD
      GO TO 1
```

```
7     CALL PRTHYD
      GO TO 1
8     CALL PUHYD
      GO TO 1
9     CALL HPLOT
      GO TO 1
10    CALL ADHYD
      GO TO 1
11    CALL SRC
      GO TO 1
12    CALL CMPRC
      GO TO 1
13    CALL STT
      GO TO 1
14    CALL CMPTT
      GO TO 1
15    CALL ROUTE
      GO TO 1
16    CALL RESVO
      GO TO 1
17    CALL ERROR
      GO TO 1
18    CALL SEDT
      GO TO 1
19    STOP
      END


      SUBROUTINE HONDO

C This subroutine reads in the data from 'datal', searches an alphanumeric
C code table to determine the NCODE of the required operation, and collects
C variables from the freefloating data field.

C The command table (CTBLE), integer table (ITBLE), number of commands
C (NCOMM) and alphanumeric array (ZALPHA) are initialized in BLOCK DATA
C located at the end of this listing.


      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20,6),
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NHD,NER,MAXNO,NCOMM,ICC,NCODE,TIME,KCODE,ICODE
      COMMON/BLOCK2/ PERQ(20,70),TQ(20,6),CC(20),LL(6),INRC,LRC

      DIMENSION CHAR(60), ALPHA(11),AUXA(10),AUXB(10)

      IF (ICC) 1,1,3
C     READ IN DATA CARD
1     READ (1,42) (ALPHA(I),I=1,11),(CHAR(I),I=1,60)
C     IF FIRST CHARACTER IS BLANK THE CARD IS A CONTINUATION OF
C     PREVIOUS CARD.
      IF (ALPHA(1)-ZALPHA(11)) 2,9,2
2     IF (ICC) 3,3,40
C     ASTERISK IN COL. 80 MEANS SKIP TO NEW PAGE BEFORE PRINTING CARD
3     IF (CHAR(60)-ZALPHA(11)) 4,5,4
```

```
4     WRITE (6,43)
5     WRITE (6,44) (ALPHA(I),I=1,11),(CHAR(I),I=1,60)
C     IF FIRST CHARACTER IS A *  THE PREVIOUS CARD WAS A COMMENT CARD
      IF (ALPHA(1)-ZALPHA(12)) 10,6,10
C     IF PUNCH CODE POSITIVE, COMMENT CARDS ARE PUNCHED.
6     IF (NPU) 8,8,7
7     WRITE (7,45) (ALPHA(I),I=1,11),(CHAR(I),I=1,60)
8     ICC=0
      GO TO 1
9     WRITE (6,44) (ALPHA(I),I=1,11),(CHAR(I),I=1,60)
      GO TO 24
C     SEARCH FIRST TWO ALPHAMERIC CHARACTERS TO SEE IF THEY ARE NUMBERS
10    ICC=1
      DO 12 I=1,10
      IF (ALPHA(1)-ZALPHA(I)) 11,15,11
11    IF (ALPHA(2)-ZALPHA(I)) 12,15,12
12    CONTINUE
C     STATEMENT NUMBER 7 IS BRANCHED TO IF NUMBERS ARE PRESENT
C     IF NOT NUMBER SEARCH COMMAND TABLE FOR MATCH
C     CALL FIRST 10 VALUES FROM PERMANENT DATA STORAGE
      DO 14 I=1,NCOMM
      DO 13 J=1,11
      IF (CTBLE(I,J)-ALPHA(J)) 14,13,14
C     SN 10=PART MATCH
13    CONTINUE
C     IF THIS LOOP IS COMPLETED WE HAVE COMPLETE MATCH- CALL NCODE
C     AND MAX NUMBER AND EXIT LOOP
      NCODE=ITBLE(I,1)
      MAXNO=ITBLE(I,2)
      GO TO 21
14    CONTINUE
C     IF MAJOR LOOPS FINISHED WITHOUT A MATCH WRITE ERROR MESSAGE
C     AND SET NER  = 1
      NER=1
      WRITE (6,46)
      RETURN
C     CONVERT DIGIT INPUT CODE FROM ALPHAMERIC TO INTEGER FORM
15    NCODE=GIT(ALPHA,1,2,1.)+0.5
C     FIND MAX NUMBER OF DATA ITEMS FOR THIS NCODE
      DO 17 I=1,NCOMM
      IF (ITBLE(I,1)-NCODE) 17,16,17
16    MAXNO=ITBLE(I,2)
      GO TO 21
17    CONTINUE
C     SEARCH DATA ROUTINE
C     SEE IF ANY DATA FOR THIS CARD
      DO 19 I=1,NCOMM
      IF (ITBLE(I,1)-NCODE) 19,18,19
18    MAXNO=ITBLE(I,2)
      GO TO 20
19    CONTINUE
20    CONTINUE
21    IF (MAXNO) 23,22,23
22    RETURN
C      ZERO ARRAYS AND COUNTERS
23    DO 47 I=1,310
47    DATA (I)=0.
      NDATA=1
```

```
24    NCHAR=0
25    DO 26 I=1,10
      AUXA(I)=0.
26    AUXB(I)=0.
      IT1=1
      IT2=1
      SIGN=1.
      LDGIT=0
      KDGIT=0
C     CARRY OUT DIGIT BY DIGIT SEARCH AND ACCUMULATION
27    NCHAR=NCHAR+1
C     HAVE WE CONSIDERED ALL CHARACTERS - RETURN IF SO
      IF (NCHAR-60) 28,32,1
28    DO 29 I=1,15
      IF (CHAR(NCHAR)-ZALPHA(I)) 29,30,29
29    CONTINUE
      GO TO 32
30    GO TO (33,33,33,33,33,33,33,33,33,33,32,27,36,32,31,27), I
C     SN 39  HANDLES SIGN CONTROL ON 1130 VERSION
31    SIGN=-1.0
      GO TO 27
C     CHARACTER IS BLANK OR COMMA - DOES IT FOLLOW A DIGIT
32    GO TO (27,48), IT1
C     CHARACTER IS A DIGIT - HAS A DECIMAL BEEN ENCOUNTERED
33    GO TO (34,35), IT2
34    LDGIT=LDGIT+1
      IT1=2
      AUXA(LDGIT)=CHAR(NCHAR)
      GO TO 27
35    KDGIT=KDGIT+1
      AUXB(KDGIT)=CHAR(NCHAR)
      GO TO 27
C     CHARACTER IS A DECIMAL - DOES IT FOLLOW A DIGIT
36    GO TO (37,38), IT1
37    IT1=2
      LDGIT=1
38    IT2=2
      GO TO 27
C     ROUTINE TO CONVERT ALPHABETIC ARRAY TO FLOATING POINT NUMBER
48    DATA (NDATA)=GIT(AUXA,1,LDGIT,1.)+GIT(AUXB,1,10,0.)
      DATA (NDATA)=DATA(NDATA)*SIGN
C     IS ALL DATA FURNISHED  YES-RETURN  NO INCREASE N DATA KEEP ON
      IF (NDATA-MAXNO) 41,39,39
39    ICC=0
40    RETURN
41    NDATA=NDATA+1
      GO TO 25
C
42    FORMAT (2A1,9A2,60A1)
43    FORMAT (1H1)
44    FORMAT (5X,2A1,9A2,60A1)
45    FORMAT (2A1,9A2,60A1)
46    FORMAT (10X,20HCOMMAND NOT IN TABLE)
      END


      FUNCTION GIT (TCARD,J,JLAST,SHIFT)
```

```
C Converts alphabetic array to floating point number

      DIMENSION TCARD(10), A(10)
      DATA A(1)/1H1/,A(2)/1H2/,A(3)/1H3/,A(4)/1H4/,A(5)/1H5/,A(6)/1H6/
      DATA A(7)/1H7/,A(8)/1H8/,A(9)/1H9/,A(10)/1H0/

      GIT=0.
      TEN=10.
      SUM=0.
      DO 3 JNOW=J,JLAST
      TTEST=TCARD(JNOW)
C     CHECK FOR LAST ENTRY
      IF (TTEST.EQ.0.) GO TO 4
C     FIND NUMBER AND COMPUTE VALUE
      DO 2 NUMB=1,10
      IF (TTEST-A(NUMB)) 2,1,2
1     ZTEST=NUMB
      IF (ZTEST.EQ.10.) ZTEST=0.
      SUM=SUM*TEN+ZTEST
      GO TO 3
2     CONTINUE
3     CONTINUE
4     IF (SHIFT) 6,5,6
5     FI=JNOW-1
      SUM=SUM*(0.1**FI)
6     GIT=SUM
      RETURN
      END


      SUBROUTINE STHYD

C     THIS SUBROUTINE STORES THE COORDINATES OF HYDROGRAPHS.
C     ADDS BASEFLOW TO HYDROGRAPH STORED

      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20,6),
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NHD,NER,MAXNO,NCOMM,ICC,NCODE,TIME,KCODE,ICODE

      DIMENSION DUMMY(300)

      ID=DATA(1)
      NHD=DATA(2)
      DT(ID)=DATA(3)
      IF(KCODE.EQ.0)GO TO 10
      DATA(4)=DATA(4)/2.590
      DATA(5)=DATA(5)/0.02832
      DO 11 J=6,305
      DATA(J)=DATA(J)/.02832
11    CONTINUE
10    DA(ID)=DATA(4)
      BSF=DATA(5)
C     BASEFLOW
```

```
      J=6
C     REMAINING DATA ARE FLOW RATES
      IF(BSF.GT.0)THEN
      OCFS(1,ID)=DATA(J)+BSF
      GOTO 51
      ENDIF
      OCFS(1,ID)=DATA(J)
51    PEAK(ID) = 1.
      RO = DATA(J)
      DO 4 I=2,300
      J=J+1
      IF(BSF.GT.0)THEN
      OCFS(I,ID)=DATA(J)+BSF
      GOTO 50
      ENDIF
      OCFS(I,ID)=DATA(J)
50    RO = RO + OCFS(I,ID)
C     IS FLOW RECEDING
      IF (OCFS(I,ID)-OCFS(I-1,ID)) 1,2,2
C     HAS FLOW RECEDED TO CUTOFF RATE
1     IF (OCFS(I,ID)) 5,5,4
C     DETERMINE PEAK FLOW
2     IF(OCFS(I,ID) - PEAK(ID)) 4,4,3
3     PEAK(ID) = OCFS(I,ID)
4     CONTINUE
5     IEND(ID)=I-1
      M=IEND(ID)
      ROIN(ID) = RO*DT(ID)*3600
      IF(NPU.LE.0)GO TO 7
      IF(ICODE.EQ.0)GO TO 6
      ROIN1=ROIN(ID)* 0.02832
      DA1=DA(ID)*2.590
      PEAK1=PEAK(ID)*.02832
      DO 13 J=1,M
      DUMMY(J)=OCFS(J,ID)*0.02832
13    CONTINUE
      WRITE(7,14)ID,NHD,DT(ID),DA1,PEAK1,ROIN1,IEND(ID),ICODE
      WRITE(7,15)(DUMMY(I),I=1,M)
      RETURN
C     PUNCH CODE
6     WRITE(7,8)ID,NHD,DT(ID),DA(ID),PEAK(ID),ROIN(ID),IEND(ID),ICODE
      WRITE (7,9) (OCFS(J,ID),J=1,M)
7     RETURN
C
8     FORMAT(   'RECALL HYD',T21,'ID=',I1,T29,'HYD NO=',I3,T42,'DT=',F9.
     &6,' HRS',T61,'DA=',F8.3,' SQ MI'/T21,'PEAK=',F7.0,'CFS',T40,'RO=',
     &F6.3," CFS ",T59,"NO PTS =",I3/T21,"CODE=",I1/T21,
     &"FLOW RATES")
9     FORMAT (T21,7F8.0)
14    FORMAT("RECALL HYD",T21,"ID=",I1,T29,"HYD NO =",I3,T42,
     &"DT=",F9.6,"HRS",T61,"DA=",F8.3,"SQ KM"/T21,"PEAK",F7.2,
     &"CMS",T40,"RO=",F6.0," CUMECS ",T59,"NO PTS=",I3/T21,"CODE=",
     &I1/T21,"FLOW RATES")
15    FORMAT (T21,7F8.2)
      END
```

```
      SUBROUTINE RECHD

C     THIS SUBROUTINE RECALLS PREVIOUSLY COMPUTED AND PUNCHED
C     HYDROGRAPHS

      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20,6),
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NHD,NER,MAXNO,NCOMM,ICC,NCODE,TIME,KCODE,ICODE
      MET1=DATA(8)
      IF(MET1.EQ.0)GO TO 2
      DATA(4)=DATA(4)/2.590
      DATA(5)=DATA(5)/.02832
      DATA(6)=DATA(6)/25.4
      M=DATA(7)
      DO 3 I=9,M+9
      DATA(I)=DATA(I)/0.02832
3     CONTINUE
2     ID=DATA(1)
      NHD=DATA(2)
      DT(ID)=DATA(3)
      DA(ID)=DATA(4)
      PEAK(ID)=DATA(5)
      ROIN(ID)=DATA(6)
      IEND(ID)=DATA(7)
      M=IEND(ID)
      J = 9
C     REMAINING DATA ARE FLOW RATES
      DO 1 I=1,M
      OCFS(I,ID)=DATA(J)
1     J=J+1
      RETURN
      END


      SUBROUTINE CMPHYD

C This subroutine develops a unit hydrograph, converts rainfall data
C into runoff by calling the soil moisture finite difference model,
C or the Curve Number routine,
C and sums these two to produce the storm runoff hydrograph.


      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20,6),
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NHD,NER,MAXNO,NCOMM,ICC,NCODE,TIME,KCODE,ICODE

      DIMENSION DUMMY(300)
      TEMP=0.


C Input data read into subroutine
```

```
      ID=DATA(1)
      NHD=DATA(2)
      DT(ID)=DATA(3)

      IF(KCODE.NE.0)THEN
C           Convert metric to imperial
            DATA(4)=DATA(4)/2.590
            IF(DATA(6).LT.0)GO TO 40
            DATA(6)=DATA(6)/0.3048
            DATA(7)=DATA(7)/1.6
      ENDIF

40    DA(ID)=DATA(4)
C
      CN=DATA(5)


C Data items 6 and 7 normally hold watershed height and length and
C from these the constants XK(recession constant) and Tp(time to peak)
C can be calculated.
C If XK and Tp are known however, they can be entered instead
C and a negative sign is put before their values.

      IF (DATA(6).LT.0.)THEN
         XK=-DATA(6)
         TP=-DATA(7)
      ELSE
         HT=DATA(6)
         XL=DATA(7)
         SLOPE=HT/XL
         XLDW=(XL**2.)/DA(ID)
         XK=27.0*(DA(ID)**.231)*(SLOPE**(-.777))*(XLDW**.124)
         TP=4.63*(DA(ID)**.422)*(SLOPE**(-.46))*(XLDW**.133)
      ENDIF



C The storm runoff array is intialised to 0, and peak of hydrograph to 1

        DO 4 I=1,300
4       OCFS(I,ID)=0.
        PEAK(ID)=1.

C Compute 'N' by iteration
      XN=5.0
      XKTP=XK/TP
        DO 6 I=1,50
        TINF=1.+SQRT(1./(XN-1.))
        XN1=.05/(XKTP*(ALOG(TINF/(TINF+.05))+.05))+1.
        DIFF=ABS(XN1-XN)
        IF (DIFF-.001) 7,7,5
5       XN=XN1
6       CONTINUE
      WRITE (6,29)
29    FORMAT(' N DID NOT CONVERGE AFTER 50 ITERATIONS.')
      GO TO 28
```

```
C Compute 'C1'
7     DELT=TINF/100.
      TC1=0.
      XN1P=XN-1.
      XN1M=1.-XN
        DO 8 I=2,101
        TC1=TC1+DELT
8       CFS(I)=(TC1**XN1P)*EXP(XN1M*(TC1-1.))
      SUM=CFS(101)/2.
        DO 9 I=2,100
9       SUM=SUM+CFS(I)
      C1=SUM*DELT
C
C Compute 'B'
      CFSII=CFS(101)
      TTINF=TINF*TP
      TREC1=TTINF+2.*XK
      EEE=EXP((TTINF-TREC1)/XK)
      XK1=3.*XK
      B=645.333/(C1+CFSII*(XKTP*(1.-EEE)+EEE*(XK1/TP)))
C
C Compute 'QP' and 'CFSI'
C
      QP=(B*DA(ID))/TP
      CFSI=QP*CFS(101)
      CFSR1=CFSI*EEE
      IF(ICODE.EQ.0)GO TO 45
      QP1=QP*.02832
      WRITE(6,38)XN,QP1
38    FORMAT(' Shape constant, N = ',F6.3/' Unit peak = ',F10.1,1X
     &,'cms'/)
      GO TO 44
45    WRITE (6,30) XN,QP
30    FORMAT(' Shape constant, N = ',F6.3/' Unit peak = ',F10.1,1X
     *,'cms'/)
C
44    CONTINUE
C
C Determine the incremental runoff
C
      IF(KCODE.NE.0)THEN
          IF(DATA(8).LT.0)GO TO 13
C             Convert rainfall data from mm to inches.
           DO 34 K=8,308
             DATA(K)=DATA(K)/25.4
34           CONTINUE
      ENDIF
C
35    J=8
      IF (DATA(J)) 13,10,10
10    RAIN(1)=DATA(J)
        DO 11 I=2,300
        J=J+1
        RAIN(I)=DATA(J)
        IF (RAIN(I)-RAIN(I-1)) 12,11,11
11      CONTINUE
12    NUMB=I-1
C
```

```
C
C     Curve number routine
13    IF(CN.LE.0)GOTO 201
C     STORAGE
      R=1000./CN-10
      B1=.2*R
      DO 15 I=1,NUMB
      IF(RAIN(I)-B1)33,33,14
33    DATA(I)=0
      Q1=0
      GOTO 15
14    Q2=((RAIN(I)-B1)**2.)/(RAIN(I)+.8*R)
      DATA(I)=Q2-Q1
      Q1=Q2
15    CONTINUE
      GOTO 202
C
C
201   DO 5555 I=1,300
5555  DATA(1)=0
      TEMP=DT(ID)
C
      CALL SOILM(TEMP,NUMB,RAIN,DATA)

C If no runoff has been generated then the simulation
C stops.

      DO 100 I=1,NUMB
      IF(DATA(I).EQ.0.)GOTO 100
      GOTO 200
100   CONTINUE
      WRITE(6,300)
300   FORMAT(' model generated no runoff'/
     &' Simulation terminates')
      STOP
200   CONTINUE

Compute unit hydrograph

202     T2=0.
        CFS(1)=0.
        DO 20 I=2,300
        T2=T2+DT(ID)
        IF (T2-TTINF) 16,16,17
16      CFS(I)=QP*((T2/TP)**XN1P)*EXP(XN1M*(T2/TP-1.))
        GO TO 20
17      IF (T2-TREC1) 18,18,19
18      CFS(I)=CFSI*EXP((TTINF-T2)/XK)
        GO TO 20
19      CFS(I)=CFSR1*EXP((TREC1-T2)/XK1)
        IF (CFS(I)-1.) 21,21,20
20      CONTINUE
      I=300
21    ICND=I
C
C
C Compute the storm runoff hydrograph by summing the unit hydrograph and
C the runoff.
```

```
C
C
      DO 24 J=2,NUMB
      N=J+ICND-2
      IF (N-300) 23,23,22
22    N=300
23    I = 2
        DO 24 K= J,N
        OCFS(K,ID)=OCFS(K,ID)+DATA(J)*CFS(I)
        I=I+1
24      CONTINUE
C
C Compute the runoff volume and determine the peak.
C
C
      RO = 0.
        DO 26 I = 2,N
        RO = RO + OCFS(I,ID)
        IF (OCFS(I,ID)-PEAK(ID))26,26,25
25      PEAK(ID)= OCFS(I,ID)
26      CONTINUE
      IEND (ID) = N
      ROIN(ID)=RO*DT(ID)*3600
C
C     PUNCH CODE
      IF (NPU) 28,28,27
27    IF(ICODE.EQ.0)GO TO 39
      ROIN1=ROIN(ID)*0.02832
      DA1=DA(ID)*2.590
      PEAK1=PEAK(ID)*.02832
        DO 41 J=1,N
      DUMMY(J)=OCFS(I,ID)*0.02832
41    CONTINUE
      WRITE(7,37)ID,NHD,DT(ID),DA1,PEAK1,ROIN1,IEND(ID),ICODE
      WRITE(7,42)(DUMMY(I),I=1,N)
      RETURN
39    WRITE(7,31)ID,NHD,DT(ID),DA(ID),PEAK(ID),ROIN(ID),IEND(ID),ICODE
      WRITE (7,32) (OCFS(I,ID),I=1,N)
28    RETURN

C
31    FORMAT(   'RECALL HYD',T21,'ID=',I1,T29,'HYD NO=',I3,T42,'DT=',F9.
     &6,' HRS',T61,'DA=',F8.3,' SQ MI'/T21,'PEAK=',F7.0,'CFS',T40,'RO=',
     &F6.3,' CFS',T59,'NO PTS=',I3/T21,"CODE=",I1/T21,'FLOW RATES')
37    FORMAT(   'RECALL HYD',T21,'ID=',I1,T29,'HYD NO=',I3,T42,'DT=',F9.
     &6,' HRS',T61,'DA=',F8.3,' SQ KM'/T21,'PEAK=',F7.2,'CMS',T40,'RO=',
     &F6.0,' CUMECS  ',T59,'NO PTS=',I3/T21,"CODE=",I1/T21,'FLOW RATES')
42    FORMAT (T21,7F8.2)
32    FORMAT (T21,7F8.0)
      END



      SUBROUTINE SOILM(DT,IR,CUMRAIN,DATA)

C A physically based parameter infiltration model which simulates near surfac
C soil water movement, and hence runoff.
```

```
C Variables used in this subroutine

C     TIME         Time when simulation begins (hours).
C     SR1          Soil water content at saturation  layer 1.
C     SR2                                 (m3/m3)  layer 2.
C     SR3                                          layer 3.
C     NLA          Number of cells in layer 1.
C     NLB          Number of cells in layer 2.
C     NL           Total number of cells in column
C     SATCON       Saturated permeability (ms-1)  layer 1.
C     SATCON2                                     layer 2.
C     SATCON3                                     layer 3.
C     EMAX         Maximum evaporation during the day (ms-1).
C     SIMDUR       Simulation duration (hours).
C     DETCAP       Surface detention capacity (m).
C     AF           Simulation iteration period (secs).
C     WT           Write-out time period (hrs).
C     THETA        Initial soil water content for each cell (m3/m3).
C     TCOM         Thickness of each cell.
C     ALR          Rain start time (hours).
C     AMR          Rain stop time.
C     NQ           Number of observations on suction moisture curve.
C     X            Moisture values....layer 1 (m3/m3).
C     Y            Suction values.....layer 1 (bars).
C     X2                              layer 2.
C     Y2                              layer 2.
C     X3                              layer 3.
C     Y3                              layer 3.
C     IR            Number of rainfall observations.
C     DT            Rainfall data time increments (hours).
C     CUMRAIN       Cumulative rainfall data at DT time increments (inches).
C     NSCOL         Number of soil columns.
C     IPCAREA       Percent area of soil column.
C     IOUT          Determines amount of output.
C                         1 - total output
C                         0 - shorter

C  Note:
C     If SR1, SR2, SR3, SATCON, SATCON2, SATCON3, DETCAP, THETA, X, X2, or X3
C     are proceeded by an 'A', then the variable type is double precision
C     rather than real.  If SR1, SR2, SR3, SATCON, SATCON2, SATCON2, DETCAP,
C     OR THETA are preceeded by an 'S',then the variable represents the
C     standard deviation of that particular soil hydrological characteristic.

C     SCURV1        Standard deviation of soil moisture curve for layer 1
C     SCURV2                                                    layer 2
C     SCURV3                                                    layer 3




C                  ---------------
C                  INITIAL SECTION
C                  ---------------
C
C
```

```
C
      DIMENSION FLUX(20),TCOM(20),SWP(20),THETA(20),COND(20)
      DIMENSION VOL(20),ANFLUX(20),AVCOND(20),DEPTH(20),DIST(20)
      DIMENSION X(20),Y(20),G(20),GZ(20),FSWP(20),CNT(20)
      DIMENSION CUMRAIN(251),Z(20),PPT(250),XP(20),FS(20)
      DIMENSION DATA(300),WDATA(300,10),HPOT(20)
      DIMENSION G2(20),Y2(20),X2(20),GZ2(20),Z2(20)
      DIMENSION G3(20),Y3(20),X3(20),GZ3(20),Z3(20)
      DIMENSION RSAT(20)
      DIMENSION AX(20),AX2(20),AX3(20),ATHETA(20)
      DIMENSION XNEW(20),YNEW(20),X2NEW(20),Y2NEW(20),
     &          X3NEW(20),Y3NEW(20)

      DOUBLE PRECISION G05DDF
      DOUBLE PRECISION DLOG10
      DOUBLE PRECISION ATHETA,AX,AX2,AX3,ADETCAP,ASR1,ASR2,ASR3,
     *   ASATCON,ASATCON2,ASATCON3,BSATCON,BSATCON2,BSATCON3,
     *   SDETCAP,SSR1,SSR2,SSR3,STHETA,SSATCON,SSATCON2,SSATCON3,
     *   SCURV1,SCURV2,SCURV3
C
C
C
C          READ IN DATA
C          ------------
C
C
C
      READ(25,*)TIME,ALR,AMR,SIMDUR
      READ(25,*)IOUT
      READ(25,*)AF,WT
      READ(25,*)NSCOL


C The array RAIN which is passed to the subroutine as a cumulative
C rainfall total is in inches.This has to be transfered to array
C PPT which is in m and represents the total for each time increment.
      IRR=IR-1
      DO 100 I=1,IRR
100   PPT(I)=(CUMRAIN(I+1)-CUMRAIN(I))*.0254


      DO 34543 W=1,NSCOL
C     For each soil column in turn, read in data and proceed through
C     simulation to determine runoff


      READ(25,*)IPCAREA
      READ(25,*)NL,NLA,NLB
      READ(25,*)(TCOM(I),I=1,NL)
      READ(25,*)EMAX,ADETCAP,SDETCAP
      READ(25,*)ASR1,SSR1,ASR2,SSR2,ASR3,SSR3
      READ(25,*)ASATCON,SSATCON,ASATCON2,SSATCON2,ASATCON3,SSATCON3
      READ(25,*)(ATHETA(I),I=1,NL)
      READ(25,*)STHETA
      READ(25,*)NQ
      READ(25,*)(AX(I),I=1,NQ)
      READ(25,*)(Y(I),I=1,NQ)
      READ(25,*)SCURV1
```

```
      READ(25,*)(AX2(I),I=1,NQ)
      READ(25,*)(Y2(I),I=1,NQ)
      READ(25,*)SCURV2
      READ(25,*)(AX3(I),I=1,NQ)
      READ(25,*)(Y3(I),I=1,NQ)
      READ(25,*)SCURV3

      NQJ=NQ
      NLL=NL+1

      IF(AMR.LT.ALR)THEN
           AMR=AMR+24.0
      ENDIF

C
C
C                CHECK DATA INPUTS
C                -----------------
C
      NERROR=0

C Check number of cells in soil column
      IF(NLA+NLB.GE.NL)THEN
         WRITE(6,1015)
1015     FORMAT(' Error-NLA,NLB,NL')
         NERROR=NERROR+1
      ENDIF
C
C Check dimensions of input vectors
      IF(NQ.GT.20.OR.NL.GT.20.OR.IR.GT.250)THEN
         WRITE(6,1020)
1020     FORMAT(' Error-limit exceeded,NQ,NL,IR')
         NERROR=NERROR+1
      ENDIF
C
C Check rainfall passed from CMPHYD
      KN=IR-1
      DO 50 I=1,KN
         IF(CUMRAIN(I+1).LT.CUMRAIN(I))THEN
            WRITE(6,1030)
1030        FORMAT(' Error-not cumulative rainfall totals')
            NERROR=NERROR+1
         ENDIF
50    CONTINUE
C
C Check that initial moisture content of each cell lies within the range of
C the suction moisture curve and does not exceed stated saturated moisture
C content.
      DO 51 I=1,NLA
         IF(ATHETA(I).GT.ASR1)THEN
            WRITE(6,1050)
1050        FORMAT(' Error-THETA larger then sat moisture content(1)')
            NERROR=NERROR+1
         ENDIF
         IF (ATHETA(I).GT.AX(NQ).OR.ATHETA(I).LT.AX(1))THEN
            WRITE(6,1055)
1055        FORMAT(' Error-THETA outside range of curves-(1)')
         ENDIF
```

```
51    CONTINUE
      NLAA=NLA+1
      NLH=NLA+NLB
      DO 52 I=NLAA,NLH
         IF(ATHETA(I).GT.ASR2)THEN
            WRITE(6,1060)
1060        FORMAT(' Error-THETA larger than sat moisture content(2)')
            NERROR=NERROR+1
         ENDIF
      IF(ATHETA(I).GT.AX2(NQ).OR.ATHETA(I).LT.AX2(1))THEN
         WRITE(6,1065)
1065     FORMAT(' Error-THETA outside range of curve-(2)')
         NERROR=NERROR+1
      ENDIF
52    CONTINUE
      NLBB=NLB+NLA+1
      DO 53 I=NLBB,NL
         IF(ATHETA(I).GT.ASR3)THEN
            WRITE(6,1070)
1070        FORMAT(' Error-THETA larger than sat moisture content(3)')
            STOP
         ENDIF
         IF(ATHETA(I).GT.AX3(NQ).OR.ATHETA(I).LT.AX3(1))THEN
            WRITE(6,1075)
1075        FORMAT(' Error-THETA outside range of curve -(2)')
            NERROR=NERROR+1
         ENDIF
53    CONTINUE
C
      IF (NERROR.NE.0)THEN
          WRITE(6,1076)NERROR
1076      FORMAT(' SOILM:   number of input data errors  ',I2,
     &'Simulation terminates')
          STOP
      ENDIF
C
C
C
C
C        DEPTH CALCULATION
C        -----------------
C
C
C
C The variable DEPTH is calculated. This refers to the distance from
C ground level to any cell midpoint.
C DIST refers to the distance between any two adjacent cell midpoints.
C
      DIST(1)=TCOM(1)/2.
      DEPTH(1)=DIST(1)
      DO 110 I=2,NL
      DEPTH(I)=DEPTH(I-1)+0.5*(TCOM(I-1)+TCOM(I))
110   DIST(I)=0.5*(TCOM(I-1)+TCOM(I))
C
C
C
C
C          PARAMETER VARIABILITY
```

```
C          --------------------
C
C  Five input variables, detention capacity, soil water content at
C saturation, soil moisture content at given tensions, saturated conductivity
C and initial moisture content are varied stochastically.
C NAG functions are called which return a 'psuedo random' value from a
C distribution with a given standard deviation and mean.
C All are assumed to have a normal distribution except the saturated
C conductivity which takes on a lognormal.
C
C
C
C Generate only one set of stochastic variables to run in HYMO.
C
C
C            RANDOM PARAMETER VALUE
C            ---------------------
C
      WRITE(6,1079)
1079    FORMAT(' INCREMENTAL RUNOFF-Parameter variability included'//)
C
C     Detention capacity.
          DETCAP=G05DDF(ADETCAP,SDETCAP)
          IF(DETCAP.LT.0.)DETCAP=0.0
          SD=SDETCAP
          WRITE(6,1180)SD
1180      FORMAT(' SD of detcap ',F5.3)
C
C     Soil water content at saturation
          SR1=G05DDF(ASR1,SSR1)
          SR2=G05DDF(ASR2,SSR2)
          SR3=G05DDF(ASR3,SSR3)
          SD1=SSR1
          SD2=SSR2
          SD3=SSR3
          WRITE(6,1181)SD1,SD2,SD3
1181      FORMAT(' SD of saturated soil content',F5.3,' layer 1'/
     &           '                            ',F5.3,' layer 2'/
     &           '                            ',F5.3,' layer 3')
C
C     Soil moisture content at given tensions
C     Layer 1
        CALL SMCURV(SR1,NQ,AX,Y,XNEW,YNEW,SCURV1)
        DO 120 I=1,20
          X(I)=XNEW(I)
120       Y(I)=YNEW(I)
C     Layer 2
        CALL SMCURV(SR2,NQ,AX2,Y2,X2NEW,Y2NEW,SCURV2)
        DO 130 I=1,20
          X2(I)=X2NEW(I)
130       Y2(I)=Y2NEW(I)
C     Layer 3
        CALL SMCURV(SR3,NQ,AX3,Y3,X3NEW,Y3NEW,SCURV3)
        DO 140 I=1,20
          X3(I)=X3NEW(I)
140       Y3(I)=Y3NEW(I)
        SD1=SCURV1
        SD2=SCURV2
```

```
      SD3=SCURV3
      WRITE(6,1182)SD1,SD2,SD3
1182  FORMAT(' SD of suction moisture curve', F5.3,' layer 1'/
     &       '                            ', F5.3,' layer 2'/
     &       '                            ',F5.3,' layer 3')
C
C Saturated conductivity for each layer
        BSATCON=DLOG10(ASATCON)
        SATCON=G05DDF(BSATCON,SSATCON)
        SATCON=10**SATCON
        BSATCON2=DLOG10(ASATCON2)
        SATCON2=G05DDF(BSATCON2,SSATCON2)
        SATCON2=10**SATCON2
        BSATCON3=DLOG10(ASATCON3)
        SATCON3=G05DDF(BSATCON3,SSATCON3)
        SATCON3=10**SATCON3
        SD1=SSATCON
        SD2=SSATCON2
        SD3=SSATCON3
        WRITE(6,1183)SD1,SD2,SD3
1183    FORMAT(' SD of sat conductivity',F5.3,' layer 1'/
     &         '                       ',F5.3,' layer 2'/
     &         '                       ',F5.3,' layer 3')
C
C Initial moisture content
        DO 150 I=1,NL
150          THETA(I)=G05DDF(ATHETA(I),STHETA)
C       Check on initial soil moisture values
        DO 160 I=1,NLA
           IF(THETA(I).GE.X(20))THETA(I)=X(20)-0.001
160        IF(THETA(I).LE.X(1))THETA(I)=X(1)+0.001
        DO 170 I=NLAA,NLH
           IF(THETA(I).GE.X2(20))THETA(I)=X2(20)-0.001
170        IF(THETA(I).LE.X2(1))THETA(I)=X2(1)+0.001
        DO 180 I=NLBB,NL
           IF(THETA(I).GE.X3(20))THETA(I)=X3(20)-0.001
180        IF(THETA(I).LE.X3(1))THETA(I)=X3(1)+0.001
      SD=STHETA
      WRITE(6,1184)SD
1184   FORMAT(' SD of initial water content',F5.3)
C
C
C
C       HYDRAULIC CONDUCTIVITY CALCULATION
C       ---------------------------------
C
C
C
C The hydraulic conductivity is calculated from suction moisture
C data  for each layer.
     NQJ=NQ
     CALL HYDCON(X,SATCON,SR1,Z,Y)
     CALL HYDCON(X2,SATCON2,SR2,Z2,Y2)
     CALL HYDCON(X3,SATCON3,SR3,Z3,Y3)
C
C
C
C
```

```
C         WRITE-OUT INITIAL CONDITIONS
C         ----------------------------
C
C
C
C Write-out suction moisture curve and generated K-values.
C
      WRITE(6,1080)
1080  FORMAT('0GENERATED K-MOISTURE CURVE'/
     &' Millington-Quirk Method'/
     &' Layer 1',26X,'Layer 2',26X,'Layer 3'/
     &3(' Moisture Suction     Unsat K    '))
      DO 175 I=1,20
175   WRITE(6,1090)X(I),Y(I),Z(I),X2(I),Y2(I),Z2(I),X3(I),Y3(I),Z3(I)
1090  FORMAT(1H ,3(F6.3,2X,F8.3,F15.12,2X))
C Write-out start conditions.
C
      WRITE(6,1100)
1100  FORMAT('0START CONDITIONS '/)
      WRITE(6,1110)TIME
1110  FORMAT(' Simulation start time',F4.1,'hrs')
      WRITE(6,1130)ALR,AMR
1130  FORMAT(' Precipitation begins at ',F4.1,2X,'and ends at ',F4.1)
      WRITE(6,1140)DT
1140  FORMAT(' Rainfall data time increment = ',F6.4,2X,'hrs')
      WRITE(6,1120)AF
1120  FORMAT(' Time increment for iteration period = ',F6.1,
     &2X,'secs'/)
      WRITE(6,1150)EMAX,DETCAP
1150  FORMAT(' Maximum evaporation during the day = ',F10.8,2X,'ms-1'/
     &' Surface detention capacity = ',F6.4,2X,'m'//)
C
C Calculate initial relative saturation of each cell in soil column
      DO 1151 I=1,NL
         IF(I.LE.NLA)RSAT(I)=THETA(I)/SR1
         IF(I.GT.NLA.AND.I.LT.NLBB)RSAT(I)=THETA(I)/SR2
         IF(I.GE.NLBB)RSAT(I)=THETA(I)/SR3
1151  CONTINUE

      WRITE(6,1152)
1152  FORMAT(' INITIAL SOIL COLUMN CONDITIONS'//)
      WRITE(6,1153)
1153  FORMAT(1X,'SAT',8X,'SAT HYD',6X,'CELL',1X,'DEPTH',
     &2X,'INITAL',2X,'REL'/
     &1H ,10X,'THETA',7X,'COND',9X,'NO',10X,'THETA',2X,'SAT'/
     &1H ,10X,'m3/m3',7X,'ms-1',14X,'m',5X,'m3/m3'/)
      WRITE(6,1154)SR1,SATCON,DEPTH(1),THETA(1),RSAT(1)
1154  FORMAT(' Layer 1  ',F7.4,1X,F15.12,3X,'1',2X,F6.4,1X,F7.4,1X,F5.3)
      IF(NLA.GT.1)THEN
         DO 1155 I=2,NLA
            WRITE(6,1156)I,DEPTH(I),THETA(I),RSAT(I)
1156        FORMAT(1H ,34X,I2,2X,F6.4,1X,F7.4,1X,F5.3)
1155     CONTINUE
      ENDIF
      WRITE(6,1157)SR2,SATCON2,NLAA,DEPTH(NLAA),THETA(NLAA),RSAT(NLAA)
1157  FORMAT(' Layer 2  ',F7.4,1X,F15.12,2X,I2,2X,F6.4,1X,F7.4,1X,F5.3)
      IF(NLB.GT.1)THEN
         DO 1158 I=NLA+2,NLH
```

```
            WRITE(6,1159)I,DEPTH(I),THETA(I),RSAT(I)
1159        FORMAT(1H ,34X,I2,2X,F6.4,1X,F7.4,1X,F5.3)
1158     CONTINUE
      ENDIF
      WRITE(6,1160)SR3,SATCON3,NLH+1,DEPTH(NLH+1),THETA(NLH+1),
     &RSAT(NLH+1)
1160  FORMAT(' Layer 3  ',F7.4,1X,F15.12,2X,I2,2X,F6.4,1X,F7.4,1X,F5.3)
      IF((NL-NLH).GT.1)THEN
         DO 1161 I=NLH+2,NL
            WRITE(6,1162)I,DEPTH(I),THETA(I),RSAT(I)
1162        FORMAT(1H ,34X,I2,2X,F6.4,1X,F7.4,1X,F5.3)
1161     CONTINUE
      ENDIF
C
C
C
C      INITIALISATION OF VARIABLES
C      ---------------------------
C
C
C
        DO 184 I=1,300
184     iwww=W
        WDATA(I,iwww)=0.0
      WATI=0.0
      MMM=2
        DO 185 I=2,NL
185     ANFLUX(I)=0.0
      CTIME=TIME*3600
      SRAIN1=0.0
      CUMDRN=0.
      CINFIL=0.
      SUMD=0.
      ICOUNT =0
      BR=AMR-ALR
      EVAPI=0.0
      SOG=THETA(1)/SR1
      RTOT=0.0
      ANFILT=0.0
      PPTT=0.0
      TG=0.0
C
C
C
C          BALANCE CHECK
C          -------------
C
C
C
C A calculation for the water balance check.
C The initial soil water content of the soil column.
C
      DO 190 I=1,NL
190   WATI=TCOM(I)*THETA(I)+WATI
C
C
C
C            CURVE GRADIENTS
```

```
C           ---------------
C
C
C
C Calculations of the gradients of the suction-moisture curve and the
C K-moisture curve for each layer.
C
      CALL GRAD(G,GZ,Y,X,Z)
      CALL GRAD(G2,GZ2,Y2,X2,Z2)
      CALL GRAD(G3,GZ3,Y3,X3,Z3)
C
C
C
C                ----------------------------
C                DYNAMIC SECTION - SIMULATION
C                ----------------------------
C
C
C This loop is completed for each time increment until end of simulation.
C
      ITMAX=SIMDUR*3600/AF
        DO 9995 II=1,ITMAX
      ICOUNT=ICOUNT+AF
      TG=TG+AF
      T=II
C
C
C       CALCULATE WATER VOLUME OF EACH CELL
C       -----------------------------------
C
C
      DO 200 I=1,NL
200   VOL(I)=TCOM(I)*THETA(I)
C
C
C
C
C                 24-HOUR CLOCK
C                 -------------
C
C
C Calculate REAL TIME for current iteration period using the 24-hour clock
C
      CTIME=CTIME+AF
      IF (CTIME.GE.86400)THEN
         CTIME=CTIME-86400
      ENDIF
C
C
C       SWP,HPOT,COND CALCULATIONS
C       --------------------------
C
C
C
C Calculate the soil water pressure, hydraulic potential and conductivity
C for each cell as conditions change during the simulation.
C
      CALL TWO(1,NLA,THETA,X,SWP,Y,G,HPOT,DEPTH,GZ,COND,Z)
```

```
      CALL TWO(NLAA,NLB,THETA,X2,SWP,Y2,G2,HPOT,DEPTH,GZ2,COND,Z2)
      CALL TWO(NLBB,NL,THETA,X3,SWP,Y3,G3,HPOT,DEPTH,GZ3,COND,Z3)
C
C
C
C
C          DETERMINE RAINFALL
C          ------------------
C
C
C
C Determine rainfall per second at end of the current iteration
C period.
C T1 is the time in hours when the current iteration period ends.
C Check that T1 is between the rain start and stop.
C If it is, decide which element of PPT array the data is to be taken from
C and make SRAIN equal to that precipitation per second.
C If it is not within the storm period, set SRAIN to 0.
C
C
      T1=T*AF/3600.0
      IF(T1.LE.(ALR-TIME).OR.T1.GT.(AMR-TIME))THEN
       SRAIN=0.0
      ELSE
       T2=T1-(AF/3600.)
       IELEM=((T2-(ALR-TIME))/DT)+1
       SRAIN=PPT(IELEM)/(DT*3600.0)
      ENDIF
c
c
C Increment precipitation total by amount of precipitation in current
C iteration period.
C
      PPTT=PPTT+(SRAIN*AF)
C
C
C
C          AVERAGE HYDRAULIC CONDUCTIVITY
C          ------------------------------
C
C
C
C Average hyraulic conductivity for flow through boundary between
C adjoining cells is weighted according to its thickness.
C
        DO 210 I=2,NL
210    AVCOND(I)=(COND(I-1)*TCOM(I-1)+COND(I)*TCOM(I))
     &/(TCOM(I-1)+TCOM(I))
C
C
C
C              BOTTOM BOUNDARY CONDITION
C              -------------------------
C
C
C Determine the bottom boundary condition under the assumption that
C water is flowing out of the soil column under gravity.
C
```

```
      FLUX(NLL)=COND(NL)
C
C
C
C            FLUX BETWEEN CELLS
C            ------------------
C
C
C
C The flux between each cell then follows Darcy's law in discrete form.
C
      DO 220 I=2,NL
220   FLUX(I)=(HPOT(I-1)-HPOT(I))*AVCOND(I)/DIST(I)
C
C
C
C           DETERMINE TOP BOUNDARY CONDITIONS
C           ---------------------------------
C
C
C
C Calculate the infiltration capacity.
C
      BNCAP=(0.0-HPOT(1))*0.5*(SATCON+COND(1))/DIST(1)
C
C Calculate precipitation excess
C
      IF(SRAIN1.EQ.SRAIN)THEN
        SUMD=(SRAIN-ANFILT)*AF+SUMD
      ELSE
         SUMD=0.0+SUMD
      ENDIF
      SRAIN1=SRAIN
C
C Calculate amount detained on the surface.
C
      IF(SUMD.LT.0.0)THEN
         DETAIN=0.0
      ELSE
         DETAIN=SUMD
      ENDIF
C
C Calculate evaporation, the flux into cell 1 and runoff.
C
      IF(SRAIN.GT.0.0) THEN
C
       EVAP= 0.0
C
       IF(SRAIN.LT.BNCAP.AND.DETAIN.LE.0.0)THEN
          ANFILT=SRAIN
       ELSE
          ANFILT=BNCAP
       ENDIF
       FLUX(1)=ANFILT
C
       IF(DETAIN.GT.DETCAP)THEN
         SUMD=DETCAP
         DETAIN=DETCAP
```

```
          RUNOFF=0.0
          IF(SRAIN.GT.BNCAP)RUNOFF=(SRAIN-BNCAP)*AF
          RTOT=RTOT+RUNOFF
        ELSE
          RUNOFF=0.0
        ENDIF
C
       ELSE
C
        RUNOFF=0.0
C
         IF(CTIME.GT.64800.AND.CTIME.LE.21600)THEN
            EVAP=EMAX/100.
        ELSE
            EVAP=EMAX*SIN(2.*3.14159*(CTIME-21600.)/86400.)
        ENDIF
C
        IF(DETAIN.LE.0.)THEN
            ANFILT=0.0
            FLUX(1)=EVAP*(-1.)
        ELSE
            ANFILT=BNCAP
            FLUX(1)=ANFILT
              DETAIN=DETAIN-(EVAP*AF)
        ENDIF
C
       ENDIF
C
C
C           CHANGES IN SOIL MOISTURE CONTENT
C           -------------------------------
C
C
C
      SWP(NLL)=-102.0
      DO 230 I=1,NL
C     If SWP in cell is greater then 0, it is saturated and flux must
C     therefore be 0.
      IF(SWP(I+1).GE.0.0)FLUX(I+1)=0.0
C     ANFLUX represents the net change in moisture content in the cell.
      ANFLUX(I)=FLUX(I)-FLUX(I+1)
      ANFLUX(I)=ANFLUX(I)*AF
C     Recalculate theta according to the  change influx(per unit area).
      THETA(I)=(VOL(I)+ANFLUX(I))/TCOM(I)
C     Due to recalculation, theta may be greater than possible water content
C     at saturation and therefore it is necessary to reset SWP to
C     0 and theta to the water content at saturation, the value of which is
C     entered into the model.
      IF (THETA(I).GE.SR1.AND.I.LE.NLA)SWP(I)=0.0
      IF (THETA(I).GE.SR2 AND.I.GT.NLA.AND.I.LE.NLB)SWP(I)=0.0
      IF(THETA(I).GE.SR3.AND.I.GT.NLB)SWP(I)=0.0
      IF(THETA(I).GE.SR1.AND.I.LE.NLA)THETA(I)=SR1
      IF(THETA(I).GE.SR2.AND.I.GT.NLA.AND.I.LE.NLB)THETA(I)=SR2
230   IF(THETA(I).GE.SR3.AND.I.GT.NLB)THETA(I)=SR3
C
C
C
C
```

```
C      CALCULATE CUMULATIVE TOTALS
C      ---------------------------
C
C
C
      CUMDRN=CUMDRN+FLUX(NLL)*AF
      EVAPI=EVAP*AF+EVAPI
      CINFIL=CINFIL+ANFILT*AF
      SOG=THETA(1)/SR1
C
C
C
C
C
C
C
C
C
C
C                 ---------------------------
C                 TERMINAL SECTION  WRITE OUT
C                 ---------------------------
C
C
C
C
C
C To print out data for every time increment for which PPT data is
C entered, check ICOUNT to see if that period has passed by.
      IF(ICOUNT.LT.(DT*3600)) GOTO 9995
      ICOUNT=0
C
C
C                       CALCULATE TIME FROM THE START
C                       -----------------------------
      T=T*AF/3600
      WRITE(6,1170)T
1170  FORMAT('0SOIL COLUMN CONDITIONS ',F7.3,1X,'HRS SINCE
     & SIMULATION BEGAN'/)
      IF(TG.EQ.86400.0)TG=0.0
C
C
C           WRITE-OUT CONDITIONS OF SOIL COLUMN
C           -----------------------------------
C
C
      IF(IOUT.EQ.0)GOTO 305

      WRITE(6,7780)
7780  FORMAT('  Cell  Depth    SWP    Theta    Hyd cond     Net',1X,
     &'flux    Rel sat')
        DO 300 I=1,NL
        IF(I.LE.NLA)SOG=THETA(I)/SR1
        IF(I.GT.NLA.AND.I.LT.NLBB)SOG=THETA(I)/SR2
        IF(I.GE.NLBB)SOG=THETA(I)/SR3
300     WRITE(6,1190)I,DEPTH(I),SWP(I),THETA(I),COND(I),ANFLUX(I),SOG
1190  FORMAT(I6,3F8.4,2F14.9,F9.3)
C
```

```
C
C                    WATER BALANCE CHECK
C                    -------------------
C
C Philips (1964) simple water balance;
C -------
C
C
C                                Amount added
C    (Initial soil)-(Current soil) =    by      - Evaporation- Drainage
C    ( moisture  ) ( moisture  )  infiltration       loss        loss
C
305   WATN=0.
        DO 310 I=1,NL
310     WATN=TCOM(I)*THETA(I)+WATN
      BAL=WATN-WATI-CINFIL+EVAPI+CUMDRN
      WRITE(6,1200)BAL
1200  FORMAT('0Balance check on soil column water status =',F12.7)
      BAL=(BAL*100.)/WATN
      WRITE(6,1210)BAL
1210  FORMAT(' Balance check as  column water vol.   =',F12.7,' %'/)
C
C
      IF(IOUT.EQ.0)GOTO 306

      WRITE(6,1220)EVAPI,PPTT,CINFIL,CUMDRN
1220   FORMAT(' Cumulative evaporation   = ',F12.8/
     &' Cumulative precipitation = ',F8.4/
     &' Cumulative infiltration  = ',F10.6/
     &' Cumulative drainage      = ',F10.6/)
306   IF(DETAIN.EQ.DETCAP)THEN
         WRITE(6,1222)
1222     FORMAT(' Detention capacity exceeded')
         WRITE(6,1230)RTOT,RTOT/.0254,T
1230     FORMAT(' Runoff total in the last period',F10.7,2X,'m'/
     &   ' Runoff total in the last period',F10.7,2X,'ins',
     $     F7.3/)
      ELSE
         WRITE(6,1221)DETAIN
1221     FORMAT(' Surface water = ',F10.6)
         WRITE(6,1226)
1226     FORMAT(' No runoff')
      ENDIF
C
C
C
C                    CREATION OF ARRAY DATA
C                    ----------------------
C
C
C Runoff is recorded in array WDATA
C The runoff for each soil column is weighted according to the
C percentage area which it occupies in the catchment area
      iwww=W
      WDATA(MMM,iwww)=(RTOT/.0254)*(IPCAREA/100.)
      RTOT=0.0
      MMM=MMM+1
9995  CONTINUE
```

```
C End of simulation of single soil column, it more than one, then return to
C to the beginning of this subroutine to repeat for next soil column

34543  CONTINUE


       DO 76567 I=1,MMM
C      Sum the weighted runoff for each soil column to derive total runoff
C      passed back to CMPHYD as DATA
          CUMDATA=0.
          DO 54345 J=1,NSCOL
               CUMDATA=WDATA(I,J)+CUMDATA
54345   CONTINUE
        DATA(I)=CUMDATA
76567 CONTINUE


       IR=MMM-1

       RETURN
       END


       SUBROUTINE HYDCON(X,SATCON,SR,Z,Y)

C This subroutine calculates hydraulic conductivity for each layer
C from the given soil moisture characteristic curve.
C Uses the Millington and Quirk method

       DIMENSION X(20),Y(20),Z(20)
       DO 845 I=1,20
       IIJ=20-I+1
       XII=X(IIJ)
       TOPS=0.
       BOTS=0.
          DO 846 J=1,20
          JF=20-J+1
          YJJ=Y(JF)
 846      BOTS=((2*J-1)*YJJ**(-2))+BOTS
       II=I
          DO 847 J=II,20
          JF=20-J+1
          YJJ=Y(JF)
  847     TOPS=((2*J+1-2*I)*YJJ**(-2))+TOPS
       JT=20-I+1
  845 Z(JT)=SATCON*(X(II)/SR)*TOPS/BOTS
       RETURN
       END


       SUBROUTINE TWO(NA,NB,THETA,X,SWP,Y,G,HPOT,DEPTH,GZ,COND,Z)

C This subroutine calculates soil water pressure, hydraulic potential
C and hydraulic conductivity for each cell as conditions change
```

```
C during simulation.

      DIMENSION THETA(20),X(20),SWP(20),Y(20),G(20),HPOT(20),
     &DEPTH(20),GZ(20),COND(20),Z(20)
      DO 15 I=NA,NB
         DO 16 J=1,19
         IF(THETA(I).GE.X(J).AND.THETA(I).LT.X(J+1))SWP(I)=Y(J)+G(J)*
     &   (THETA(I)-X(J))
 16      CONTINUE
      HPOT(I)=SWP(I)-DEPTH(I)
         DO 17 J=1,19
         IF(THETA(I).GT.X(J).AND.THETA(I).LE.X(J+1))COND(I)=Z(J)+GZ(J)*
     &   (THETA(I)-X(J))
17       CONTINUE
15    CONTINUE
        RETURN
        END


      SUBROUTINE GRAD(G,GZ,Y,X,Z)

C This subroutine calculates the gradients of the suction-moisture
C and hydraulic conductivity-moisture curves.
C
      DIMENSION G(20),GZ(20),Y(20),X(20),Z(20)
      DO 261 I=1,19
      G(I)=(Y(I+1)-Y(I))/(X(I+1)-X(I))
261   GZ(I)=(Z(I+1)-Z(I))/(X(I+1)-X(I))
      RETURN
      END


      SUBROUTINE SMCURV(SR,NQ,AX,Y,XNEW,YNEW,SCURV)

C  Generates a stochastic suction moisture curve to be fed into
C  soil moisture model
c
C
C
      DOUBLE PRECISION G05DDF
      DOUBLE PRECISION AX,SCURV
      DIMENSION AX(20),X(20),XNEW(20),YNEW(20),G(20),Y(20)
C
C
C  Determine the stochastic values of moisture
C
      X(1)=G05DDF(AX(1),SCURV)
      IF(X(1).LT.0.)X(1)=0.001
C
      DO 100 I=2,NQ
      X(I)=G05DDF(AX(I),SCURV)
100   IF(X(I).LE.X(I-1))X(I)=X(I-1)+0.001
      IF(X(NQ).GE.SR)SR=X(NQ)+0.001
C
C Calculate gradients of this new suction-moisture curve
c
```

```
      NNQ=NQ-1
      DO 200 I=1,NNQ
200   G(I)=(Y(I+1)-Y(I))/(X(I+1)-X(I))
C
C Calculate max and min moisture values, and determine the size of
C equal intervals.
C
      XMAX=RMAX(X,NQ)
      XMIN=RMIN(X,NQ)
      XINT=(XMAX-XMIN)/19.
C
C Determine the new values of moisture-equal intervals
C
      XNEW(1)=XMIN
      DO 300 I=2,19
300   XNEW(I)=XNEW(1)+(XINT*(I-1))
      XNEW(20)=XMAX
C
C Determine the associated new values of suction
C
      DO 350 I=1,19
        DO 400 J=1,NNQ
          IF(XNEW(I).GE.X(J).AND.XNEW(I).LT.X(J+1))
     &    YNEW(I)=Y(J)+G(J)*(XNEW(I)-X(J))
400     CONTINUE
350   CONTINUE
      YNEW(20)=Y(NQ)
C
C
C
C
      RETURN
      END



      FUNCTION RMAX (X,NQ)

C Determines the maximum real in an array

      DIMENSION X(NQ)
C
      RMAX=X(1)
      DO 10 I=2,NQ
10      IF(X(I).GT.RMAX)RMAX=X(I)
C
      RETURN
      END



      FUNCTION RMIN(X,NQ)

C Determines minimum real in an array

      DIMENSION X(NQ)
C
      RMIN=X(1)
```

```
      DO 10 I=2,NQ
10      IF(X(I).LT.RMIN)RMIN=X(I)
C
      RETURN
      END



      SUBROUTINE PRTHYD
C     CONVERTS  Q HYDROGRAPH TO  STAGE HYDROGRAPH FOR
C     SPECIFIED CROSS SECTION
C     ID=Q HYD INPUT
C     IDR=CROSS SECTION ID
C     NPK=2 OR GREATOR FOR CONVERSION Q/STAGE
C     NPK=1 Q HYD
C     NPK=0 Q PEAK AND VOLUME ONLY
C     IN = FORMAT OF OUTPUT
C     IN =0  REGULAR FORMAT
C     IN=1 PRINT DISCHARGE ONLY IN SINGLE ENTRY PER LINE
C     THIS SUBROUTINE PRINTS THE COORDINATES OF A HYDROGRAPH.

      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20 
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NBD,NER,MAXNO,NCOMM,ICC,NCODE,TIME,KCODE,ICODE
      COMMON/BLOCK2/ PERQ(20,70),TQ(20,6),CC(20),LL(6),INRC,LRC
      DIMENSION DUMMY(300),S(300,6),PEAKS
      DIMENSION ISG(6)
C New variables used
C  S  stage equivalent of OCFS
C  PEAKS peak stage (equivalent of PEAK)

C Input data is read into the subroutine.

      ID=DATA(1)
      NPK=DATA(2)
      IDR=DATA(3)
      IN=DATA(4)
      M=IEND(ID)
      WRITE(6,40)ID,NPK
      TIME1=0
      IF(NPK.LT.1)GOTO 32
      IF(NPK.LT.2)GOTO 2
C     CONVERSION TO STAGE HYDROGRAPH
C     CHECK RATING CURVE ENTERED
      IF(IDR.EQ.0)THEN
      WRITE(6,*)'NEED TO ENTER RATING CURVE ID'
      RETURN
      ENDIF
C     CHECK IF MULTIPLE ROUTING INVOKED
      IF(IDR.GT.6)GOTO 51
      IF(TQ(10,IDR).GT.0)THEN
      DO 50 I=1,20
50    Q(I,IDR)=TQ(I,IDR)
      ENDIF
      JJ=IDR
      GOTO 7
```

```
C     SEGMENT HYDROGRAPH
51    JJ=IDR/10
7     DO 3 I=1,M
      J=1
6     IF(OCFS(I,ID).LE.Q(J,IDR))GOTO 4
      J=J+1
      IF(J.GT.20)THEN
      WRITE(6,*)'RATING CURVE EXCEEDED, STOPPED'
      RETURN
      ENDIF
      GOTO 6
4     IF(OCFS(I,ID).EQ.Q(J,IDR))THEN
      S(I,ID)=C(J,JJ)
      GOTO 3
      ENDIF
C     INTERPOLATE
      S(I,ID)=C(J-1,JJ)+(((OCFS(I,ID)-Q(J-1,IDR))*(C(J,JJ)-
     &C(J-1,JJ)))/(Q(J,IDR)-Q(J-1,IDR)))
3     CONTINUE
C     TIME ARRAY
2     DO 8 I=1,M
      DATA(I)=TIME1
8     TIME1=TIME1+DT(ID)
      J=0
      M4=M+4
      M5=M4/5
      IF(NPK.LT.2)GOTO 27
      IF(ICODE.EQ.0)THEN
      WRITE(6,9)
      GOTO 10
      ENDIF
      WRITE(6,11)
10    IF(IN.GT.0)THEN
      IF(ICODE.EQ.0)THEN
      DO 38 I=1,M
38    WRITE(6,28)S(I,ID)
      RETURN
      ENDIF
      DO 43 I=1,M
      S(I,ID)=S(I,ID)*0.3048
43    WRITE(6,28)S(I,ID)
      RETURN
      ENDIF
      IF(ICODE.GT.0)THEN
      DO 45 I=1,M
45    S(I,ID)=S(I,ID)*0.3048
      ENDIF
39    J=J+1
      WRITE(6,30)(DATA(I),S(I,ID),I=J,M,M5)
      IF(J-M5)39,13,13
13    ROIN1=ROIN(ID)
      DO 16 I=1,20
      IF(Q(I,IDR)-PEAK(ID))16,17,17
16    CONTINUE
17    IF(Q(I,IDR).EQ.PEAK(ID))THEN
      PEAKS=C(I,JJ)
      GOTO 18
      ENDIF
```

```
      PEAKS=PEAK(ID)*((C(I-1,JJ)-C(I,JJ))/(Q(I-1,IDR)
     &-Q(I,IDR)))+C(I-1,JJ)-((Q(I-1,IDR)*(C(I-1,JJ)-
     &C(I,JJ)))/(Q(I-1,IDR)-Q(I,IDR)))
18    IF(ICODE.EQ.0)THEN
      WRITE(6,14)ROIN1,PEAKS
      RETURN
      ENDIF
      PEAKS=PEAKS*0.3048
      ROIN1=ROIN(ID)*0.0283168
      WRITE(6,15)ROIN1,PEAKS
      RETURN
C     DISCHARGE HYDROGRAPHS
27    IF(ICODE.EQ.1)THEN
C     METRIC
      WRITE(6,21)
      DO 23 I=1,M
23    DUMMY(I)=OCFS(I,ID)*0.0283168
      PEAK1=PEAK(ID)*0.0283168
      ROIN1=ROIN(ID)*0.0283168
      GOTO 20
      ENDIF
C     IMPERIAL
19    WRITE(6,25)
      DO 26 I=1,M
26    DUMMY(I)=OCFS(I,ID)
      PEAK1=PEAK(ID)
      ROIN1=ROIN(ID)
20    IF(IN.GT.0)THEN
      DO 29 I=1,M
29    WRITE(6,28)DUMMY(I)
      RETURN
      ENDIF
31    J=J+1
      WRITE(6,30)(DATA(I),DUMMY(I),I=J,M,M5)
      IF(J-M5)31,32,32
32    IF(ICODE.NE.0)GOTO 34
      ROIN1=ROIN(ID)
      PEAK1=PEAK(ID)
      WRITE(6,35)ROIN1,PEAK1
      RETURN
34    ROIN1=ROIN(ID)*0.0283168
      PEAK1=PEAK(ID)*0.0283168
      WRITE(6,36)ROIN1,PEAK1
      RETURN

21    FORMAT(10X,"TIME",6X," FLOW",11X,"TIME",6X," FLOW",11X,"TIME",
     &6X,"FLOW",11X,"TIME",6X,"FLOW",11X,"TIME",6X,"FLOW"/11X,"HRS",
     &7X," MS",12X,"HRS",7X," MS",12X,"HRS",7X," MS",12X,"HRS",
     &7X," MS",12X,"HRS",7X," MS")
25    FORMAT(10X,"TIME",6X," FLOW",11X,"TIME",6X," FLOW",11X,"TIME",
     &6X,"FLOW",11X,"TIME",6X,"FLOW",11X,"TIME",6X,"FLOW"/11X,"HRS",
     &7X," CFS ",10X,"HRS",7X," CFS ",10X,"HRS",7X," CFS ",10X,"HRS",
     &7X," CFS ",10X,"HRS",7X," CFS ")
30    FORMAT (5(5X,F10.3,F10.3))
40    FORMAT(  'PRINT HYD',T21,'ID=',I1,T29,'NPK=',I1)
36    FORMAT(1H0,9X,"HYDROGRAPH VOLUME=",F20.0," CUMEC  "/10X,"PEAK
     & DISCHARGE RATE =",F10.0,"CMS"///)
35    FORMAT(1H0,9X,"HYDROGRAPH VOLUME=",F20.0," CF  "/10X,"PEAK
```

```
     & DISCHARGE  RATE=",F10.0,"CFS"///)
14    FORMAT(1H0,9X,"HYDROGRAPH VOLUME=",F20.0," CF  "/10X,"PEAK
     & ELEVATION       =",F10.0," FEET"///)
15    FORMAT(1H0,9X,"HYDROGRAPH VOLUME=",F20.0,"CUMECS"/10X,"PEAK
     & ELEVATION       =",F10.0,"METRES"///)
11    FORMAT(10X,"TIME",6X,"ELEV",11X,"TIME",6X,"ELEV",11X,"TIME",
     & 6X,"ELEV",11X,"TIME",6X,"ELEV",11X,"TIME",6X,"ELEV",/11X,"HRS",
     & 7X,"M ",12X,"HRS",7X,"M ",12X,"HRS",7X,"M ",12X,"HRS",
     & 7X,"M ",12X,"HRS",7X,"M ")
9     FORMAT(10X,"TIME",6X,"ELEV",11X,"TIME",6X,"ELEV",11X,"TIME",
     & 6X,"ELEV",11X,"TIME",6X,"ELEV",11X,"TIME",6X,"ELEV",/11X,"HRS",
     & 7X,"FT",12X,"HRS",7X,"FT",12X,"HRS",7X,"FT",12X,"HRS",
     & 7X,"FT",12X,"HRS",7X,"FT")
28    FORMAT(F10.3)
      END


      SUBROUTINE PUHYD

C     THIS SUBROUTINE PUNCHES HYDROGRAPHS IN FORM TO BE USED BY
C     SUBROUTINE RECHD

      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20,6),
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NHD,NER,MAXNO,NCOMM,ICC,NCODE,TIME,KCODE,ICODE
      DIMENSION DUMMY(300)
      ID=DATA(1)
      M=IEND(ID)
      IF(ICODE.EQ.0)GO TO 3
      DA1=DA(ID)*2.590
      PEAK1=PEAK(ID)*0.02832
      ROIN1=ROIN(ID)*0.02832
        DO 4 I=1,M
      DUMMY(I)=OCFS(I,ID)*0.02832
4     CONTINUE
      WRITE(7,5)ID,NHD,DT(ID),DA1,PEAK1,ROIN1,IEND(ID),ICODE
      WRITE(7,6)(DUMMY(I),I=1,M)
      RETURN
3     WRITE(7,1)ID,NHD,DT(ID),DA(ID),PEAK(ID),ROIN(ID),IEND(ID),ICODE
      WRITE (7,2) (OCFS(I,ID),I=1,M)
      RETURN
C
1     FORMAT(   'RECALL HYD',T21,'ID=',I1,T29,'HYD NO=',I3,T42,'DT=',F9.
     &6,' HRS',T61,'DA=',F8.3,' SQ MI'/T21,'PEAK=',F7.0,'CFS',T40,'RO=',
     &F20.0," CF  ",T60,"NO PTS=",I3/21X,"CODE=",I1/T21,
     &"FLOW RATES")
5     FORMAT(   'RECALL HYD',T21,'ID=',I1,T29,'HYD NO=',I3,T42,'DT=',F9.
     &6,' HRS',T61,'DA=',F8.3,' SQ KM'/T21,'PEAK=',F7.2,'CMS',T40,'RO=',
     &F20.0," CUMEC ",T60,"NO PTS=",I3/21X,"CODE=",I1/T21,
     &"FLOW RATES")
2     FORMAT (T21,7F8.0)
6     FORMAT (T21,7F8.2)
      END
```

```
      SUBROUTINE HPLOT
C     THIS SUBROUTINE PLOTS EITHER 1 OR 2 HYDROGRAPHS ON A SET OF AXIS
      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20,6),
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NHD,NER,MAXNO,NCOMM,ICC,NCODE,TIME,KCODE,ICODE
      ID1=DATA(1)
      ID2=DATA(2)
      DATA ZERO, PLUS, BLANK, DASH, DOT/'0','+',' ','-','.'/
      MAX=121
      J=1
C     ARE THERE 1 OR 2 HYDROGRAPHS
      IF (ID2) 1,1,2
C     DETERMINE HIGHEST PEAK IF 2 HYDROGRAPHS
1     QMAX=PEAK(ID1)
      GO TO 14
2     IF (PEAK(ID1)-PEAK(ID2)) 3,3,4
3     QMAX=PEAK(ID2)
      GO TO 5
4     QMAX=PEAK(ID1)
C     IF 2 HYDROGRAPHS DETERMINE LARGEST DT AND INTERPOLATE OTHER
C     HYDROGRAPH IF NECESSARY
5     IF (DT(ID1)-DT(ID2)) 6,13,7
6     L=ID1
      K=ID2
      GO TO 8
7     L=ID2
      K=ID1
8     M=IEND(L)
      TID=DT(K)
      TIDH=0.
      DO 11 I=2,M
      TIDH=TIDH+DT(L)
      IF (TID-TIDH) 10,9,11
9     J=J+1
      CFS(J)=OCFS(I,L)
      TID=TID+DT(K)
      GO TO 11
10    J=J+1
      CFS(J)=OCFS(I-1,L)+((TID-TIDH+DT(L))/DT(L))*(OCFS(I,L)-OCFS(I-1,L)
     &)
      TID=TID+DT(K)
11    CONTINUE
      IEND(L)=J
      DT(L)=DT(K)
      DO 12 I=2,J
12    OCFS(I,L)=CFS(I)
13    IF (IEND(ID1)-IEND(ID2)) 14,14,15
14    M=IEND(ID1)
      GO TO 16
15    M=IEND(ID2)
16    XM = M
C     DETERMINE TIME SCALE
      XSCL = XM / 120
```

```
      YSCL=QMAX/50.
C     PLOT HYDROGRAPHS
      DO 20 I=1,MAX
20    CFS(I)=DASH
      IF(ICODE.EQ.0)GO TO 49
      WRITE(6,50)
50    FORMAT(T2,"FLOW RATE (CMS)")
      QMAX1=QMAX*0.02832
      WRITE(6,41)QMAX1,DOT,(CFS(I),I=1,MAX),DOT
      GO TO 51
49    WRITE(6,48)
48    FORMAT(T2,'FLOW RATE (CFS)')
      WRITE(6,41)QMAX,DOT,(CFS(I),I=1,MAX),DOT
51    Q1=QMAX
      J1=10
      DO 37 J=1,50
      IF (J-J1) 23,21,23
21    DO 22 I=1,MAX
22    CFS(I)=DASH
      GO TO 25
23    DO 24 I=1,MAX
24    CFS(I)=BLANK
25    Q2=Q1-YSCL
      DO 28 I=2,M
      IF (OCFS(I,ID1)-Q1) 26,27,28
26    IF (OCFS(I,ID1)-Q2) 28,28,27
27    XI = I
      K = XI / XSCL + 1.
      CFS(K)=ZERO
28    CONTINUE
      WRITE (6,44) DOT,(CFS(I),I=1,MAX),DOT
      IF (ID2) 34,34,29
29    DO 18 I = 1, MAX
18    CFS(I) = BLANK
      DO 33 I=1,M
      IF (OCFS(I,ID2)-Q1) 30,31,33
30    IF (OCFS(I,ID2)-Q2) 33,33,31
31    XI = I
      K = XI / XSCL + 1.
      CFS(K)=PLUS
33    CONTINUE
      WRITE (6,42) (CFS(I),I=1,MAX)
34    IF (J-J1) 36,35,36
35    J1=J1+10
      IF(ICODE.EQ.0)GO TO 52
      QD=Q2*0.02832
      WRITE(6,43)QD
      GO TO 36
52    WRITE(6,43)Q2
36    Q1=Q2
37    CONTINUE
      CFS(1)=TIME
      DTT=DT(ID1)*(XM - 1.) / 12.
C     PUT TIME ARRAY IN CFS AND WRITE TIME SCALE
      DO 38 I=2,13
38    CFS(I)=CFS(I-1)+DTT
      WRITE (6,45) (CFS(I),I=1,13)
      WRITE (6,46)
```

```
      IF (NPU) 40,40,39
 39   WRITE (7,47) ID1,ID2
 40   RETURN
C
 41   FORMAT(1X,F7.0,123A1)
 42   FORMAT(1H+,8X,121A1)
 43   FORMAT (1H+,F7.0)
 44   FORMAT(8X,123A1)
 45   FORMAT(T3,13F10.2)
 46   FORMAT(49X,'TIME HOURS'///)
 47   FORMAT(   'PLOT HYD',T21,'ID I=',I1,T29,'ID II=',I1)
      END



      SUBROUTINE ADHYD

C     THIS SUBROUTINE ADDS TWO HYDROGRAPHS.
C
C     INCLUDES CORRECTION
C     IF DT(ID1) AND DT(ID2) ARE DIFFERENT "AND"
C     ID IS EITHER EQUAL TO ID1 OR ID2 THEN
C     EXSISTING CODE WOULD HAVE LOST ITERATION PERIOD OF ONE OF
C     THE INFLOW HYDS - THIS IS NOW CORRECTED

      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20,6),
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NHD,NER,MAXNO,NCOMM,ICC,NCODE,TIME,KCODE,ICODE

      ID=DATA(1)
      NHD=DATA(2)
      ID1=DATA(3)
      ID2=DATA(4)
      KK=0
C     CHECK ARRAYS ARE NOT EMPTY
      IF(IEND(ID1).EQ.0.OR.IEND(ID2).EQ.0)THEN
      WRITE(6,*)'ONE HYDROGRAPH BEING ADDED IS ZERO'
      IF(IEND(ID1).EQ.0)THEN
      K=ID2
      GOTO 53
      ENDIF
      K=ID1
 53   DO 52 I=1,IEND(K)
 52   OCFS(I,ID)=OCFS(I,K)
      PEAK(ID)=PEAK(K)
      ROIN(ID)=ROIN(K)
      DA(ID)=DA(K)
      IEND(ID)=IEND(K)
      DT(ID)=DT(K)
      GOTO 27
      ENDIF
      IF(DT(ID1).EQ.DT(ID2))GOTO 54
      IF(ID.NE.ID1.AND.ID.NE.ID2)GOTO 54
C     DANGER OF CONFUSION IN DT. ALTER ID TO KK
 55   DO 56 KK=1,6
```

```
      IF(KK.EQ.ID1)GOTO 56
      IF(KK.EQ.ID2)GOTO 56
      GOTO 57
56    CONTINUE
57    ID=KK
54    PEAK(ID) = 1.
C     MAKE TIME INCREMENTS EQUAL IF NOT EQUAL. USE SMALLER INCREMENT
      IF (DT(ID1)-DT(ID2)) 1,3,2
1     DT(ID)=DT(ID1)
      L=ID1
      K=ID2
      GO TO 6
2     DT(ID)=DT(ID2)
      L=ID2
      K=ID1
      GO TO 6
3     DT(ID)=DT(ID1)
      IF (IEND(ID1)-IEND(ID2)) 4,4,5
4     M3=IEND(ID1)
      K1=ID2
      IEND(ID)=IEND(ID2)
      GO TO 18
5     M3=IEND(ID2)
      K1=ID1
      IEND(ID)=IEND(ID1)
      GO TO 18
C     DETERMINE DURATIONS OF FLOW
6     XIEND1=IEND(ID1)-1
      XIEND2=IEND(ID2)-1
      DUR1=XIEND1*DT(ID1)
      DUR2=XIEND2*DT(ID2)
      IF (DUR1-DUR2) 7,8,8
7     IEND(ID)=DUR2/DT(ID)+1.
      M3=DUR1/DT(ID)+1.
      K1=ID2
      GO TO 9
8     IEND(ID)=DUR1/DT(ID)+1.
      M3=DUR2/DT(ID)+1.
      K1=ID1
9     IF (IEND(ID)-300) 11,11,10
10    IEND(ID)=300
11    M2=IEND(K)
      J=1
C     INTERPOLATE ONE HYDROGRAPH IF NECESSARY
      TIDH=0.
      TID=DT(ID)
      DO 15 I=2,M2
      TIDH=TIDH+DT(K)
12    IF (TIDH-TID) 15,13,14
13    J=J+1
      DATA (J)=OCFS(I,K)
      TID=TID+DT(ID)
      IF (J-300) 15,16,16
14    J=J+1
      DATA (J)=OCFS(I-1,K)+((TID-TIDH+DT(K))/DT(K))*(OCFS(I,K)-OCFS(I-1,
     &K))
      TID=TID+DT(ID)
      IF (J-300) 12,16,16
```

```
15    CONTINUE
16    IEND(K)=J
      DO 17 I=2,J
17    OCFS(I,K)=DATA(I)
18    M=IEND(ID)
      RO = 0.
C     ADD HYDROGRAPHS
C     CONVERT KK TO ID
      IF(KK.GT.0)THEN
      ID=DATA(1)
      DT(ID)=DT(L)
      ENDIF
      DO 20 I=1,M3
      OCFS(I,ID)=OCFS(I,ID1)+OCFS(I,ID2)
      IF (OCFS(I,ID) - PEAK(ID)) 20,20,19
19    PEAK(ID) = OCFS(I,ID)
20    RO = RO + OCFS(I,ID)
      DA(ID)=DA(ID1)+DA(ID2)
      IF (PEAK(ID) - PEAK(K1)) 21,22,22
21    PEAK(ID) = PEAK(K1)
22    IF (M-M3) 25,25,23
23    M3 = M3 + 1
      DO 24 I = M3,M
      OCFS(I,ID) = OCFS (I,K1)
24    RO = RO + OCFS(I,ID)
25    ROIN(ID) =RO * DT(ID)*3600
      IF (NPU) 27,27,26
26    WRITE (7,28) ID,NHD,ID1,ID2
27    RETURN
C
28    FORMAT(   'ADD HYD',T21,'ID=',I1,T29,' HYD NO=',I3,T45,'ID I=',I1,
     &T60,'ID II=',I1)
      END


      SUBROUTINE SRC

C     THIS SUBROUTINE STORES AN ELEVATION - END AREA - FLOW TABLE.

      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20,6),
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NHD,NER,MAXNO,NCOMM,ICC,NCODE,TIME,KCODE,ICODE
      ID=DATA(1)
      VS=DATA(2)
C     VALLEY SECTION NUMBER
C     REMAINING DATA ARE ELEVATION, AREA, AND FLOW FOR EACH POINT OF
C     THE RATING CURVE
      IF(KCODE.EQ.0)GO TO 2
      J=3
      DO 3 I=1,20
      DATA(J)=DATA(J)/0.3048
      DATA(J+1)=DATA(J+1)/0.093
      DATA(J+2)=DATA(J+2)/0.02832
      J=J+3
3     CONTINUE
```

```
2     EMIN=DATA(3)
      J=3
      DO 1 I=1,20
      DEEP(I,ID)=DATA(J)-EMIN
      A(I,ID)=DATA(J+1)
      Q(I,ID)=DATA(J+2)
      J=J+3
1     CONTINUE
      RETURN
      END

      SUBROUTINE CMPRC

C     THIS SUBROUTINE COMPUTES THE DISCHARGE END-AREA ELEVATION
C     RELATIONSHIP FOR A VALLEY SECTION.
C
C     IF MUTIPLE ROUTING INVOKED -
C     COMPUTES SEPARATE RATING CURVES FOR EACH SEGMENT
C     ALSO COMPUTES % FLOW AT EACH ELEVATION FOR SEPARATE SEGMENTS
C
C     IF TURBULENT EXCHANGE INVOKED
C     COMPUTES THE RATING CURVE USING REDEFINED AREA AND WETTED
C     PERIMETER CALCULATION - KNIGHT TECHNIQUE
C     FOUR OPTIONS
C
C     NOTE --- TURBULENT EXCHANGE REDEFINITIONS USED "ONLY "
C     FOR OUT-OF-BANK ELEVATIONS
C
C     MULTIPLE ROUTING AND TURBULENT EXCHANGE OPERATES INDEPENDANTLY

      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20,6),
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NHD,NER,MAXNO,NCOMM,ICC,NCODE,TIME,KCODE,ICODE
      DIMENSION MMM(6),W(6),XM(70)
      COMMON/BLOCK2/ PERQ(20,70),TQ(20,6),CC(20),LL(6),INRC,LRC
C     new variables used
C     MMM  used to mark segments for turbulent exchange
C     1=left floodplain  2=channel  3=right floodplain
C     W width of channel segment
C     H channel depth
C     XM min elev in segment
C
      ID=DATA(1)
C     STORAGE LOCATION NUMBER. (1-6)
      IT=DATA(2)
C     TURBULENT EXCHANGE INCLUSION
      MR=DATA(3)
C     MULTIPLE ROUTING INCLUSION
      VS=DATA(4)
C     VALLEY SECTION IDENTIFICATION NUMBER.
      NSEG=DATA(5)
C     NUMBER OF SEGMENTS IN THE VALLEY SECTION.
      IF(KCODE.EQ.0)GO TO 26
      DATA(6)=DATA(6)/0.3048
      DATA(7)=DATA(7)/0.3048
25    ELO=DATA(6)
```

```
      EMAX=DATA(7)
C     MAXIMUM ELEVATION FOR COMPUTATIONS.
      SLOPE1=DATA(8)
C     CHANNEL SLOPE.
      SLOPE2=DATA(9)
C     FLOODPLAIN SLOPE.
      DIF=(EMAX-ELO)/19.
      C(1,ID)=ELO
      DO 1 I=2,20
1     C(I,ID)=C((I-1),ID)+DIF
C     SET AREA AND DISCHARGE ARRAYS = 0.
      IF(MR.GT.0) GOTO 53
      DO 2 I=1,20
      A(I,ID)=0.
2     Q(I,ID)=0.
      DO 76 K=1,NSEG
      W(K)=0
76    MMM(K)=0
      GOTO 54
53    DO 55 J=10*ID+1,10*ID+NSEG
      DO 56 I=1,20
      A(I,J)=0
      Q(I,J)=0
      TQ(I,ID)=0
56    PERQ(I,J)=0
55    CONTINUE
54    J=10
C     READ N VALUES AND SEGMENT BORDER POINTS.
      DO 3 I=1,NSEG
      SEGN(I)=DATA(J)
      IF(KCODE.NE.0)DATA(J+1)=DATA(J+1)/0.3048
      DIST(I)=DATA(J+1)
3     J=J+2
C     REMAINING DATA ITEMS ARE DISTANCES AND ELEVATIONS.
      IF(KCODE.EQ.0)GO TO 27
      DO 28 I=J,310
      DATA(I)=DATA(I)/0.3048
28    CONTINUE
27    JJJ=J
      DO 6 I=1,NSEG
4     J=J+2
      IF (DATA(J) - DIST(I)) 4,5,5
5     ISG(I) = J + 1
6     CONTINUE
C     COMPUTE CHANNEL WIDTH
      IF(IT.LT.1)GOTO 78
      J=10
35    DO 61 K=1,NSEG
      SELEV=0
      IF(SEGN(K))62,63,63
63    IF(K.EQ.1)GOTO 65
      IF(SEGN(K-1))64,65,65
65    IF(K.EQ.NSEG)GOTO 61
      IF(SEGN(K+1))66,61,61
66    IF(K.LT.2)GOTO 67
C     LEFT HAND FLOODPLAIN
      MMM(K)=1
      GOTO 61
```

```
C     CHANNEL
62    IF(K.EQ.1.OR.K.EQ.NSEG)THEN
      WRITE(6,70)
      RETURN
      ENDIF
      W(K)=(DATA(ISG(K)-1)-DATA(ISG(K-1)-1))/2
      MMM(K)=2
      GOTO 61
C     RIGHT HAND FLOODPLAIN
64    MMM(K)=3
      GOTO 61
C     LEFT HAND FLOODPLAIN
67    MMM(K)=1
61    CONTINUE
C     COMPUTE DISCHARGES AND END AREAS FOR EACH SEGMENT.
78    IF(IT.NE.0)THEN
      WRITE(6,170)IT
      ENDIF
      DO 22 K=1,NSEG
      J=JJJ
      JJJ1=JJJ+1
      IF (SEGN(K)) 7,7,8
7     SLOPE=SLOPE1
      GO TO 9
8     SLOPE=SLOPE2
9     SLPN=1.486*SLOPE**.5
C     COMPUTE AREA AND DISCHARGE FOR SEGMENT.
      DO 21 I=2,20
      AA=0.
      P=0.
      J=JJJ-1
      DEP2=0.
10    J=J+2
      IF (J-ISG(K)) 12,12,11
11    IF(AA-.001)21,21,20
12    IF(DATA(J)-C(I,ID)) 13,10,10
13    DEP1=C(I,ID)-DATA(J)
      IF (J-JJJ1) 16,16,14
14    XL=DATA(J-1)-DATA(J-3)
      DEP3=ABS(DATA(J-2)-DATA(J))
      XL=XL*DEP1/DEP3
15    AA=AA+XL*(DEP1+DEP2)/2.
      P=P+SQRT((DEP1-DEP2)**2+XL**2)
16    DEP2=DEP1
      J=J+2
      IF (J-ISG(K)) 17,17,20
17    IF (DATA(J)-C(I,ID)) 18,18,19
18    DEP1=C(I,ID)-DATA(J)
      XL=DATA(J-1)-DATA(J-3)
      GOTO 15
19    DEP1=0
      XL=DATA(J-1)-DATA(J-3)
      DEP3=ABS(DATA(J-2)-DATA(J))
      XL=XL*DEP2/DEP3
      AA=AA+XL*(DEP1+DEP2)/2.
      P=P+SQRT((DEP1-DEP2)**2+XL**2)
      DEP2=0.
      GO TO 10
```

```
C     CHECK IF TURBULENT EXCHANGE INVOKED
C     CHECK IF OUT-OF-BANK
20    IF(MMM(K).LT.1)GOTO 98
      IF(MMM(K).EQ.1)GOTO 90
      IF(MMM(K).EQ.3)GOTO 91
C     CHANNEL
C     CHECK OUT-OF-BANK
      IF(C(I,ID).LE.DATA(ISG(K)).AND.C(I,
     &ID).LE.DATA(ISG(K-1)))GOTO 98
      H=(DATA(ISG(K))+DATA(ISG(K-1)))/2-ELO
      IF(IT.LE.2)GOTO 92
C     AREA METHOD 3 AND 4
      AA=AA/2+(W(K)*H)
92    IF(IT.EQ.1.OR.IT.EQ.3)THEN
C     WETTED PERIMETER METHOD 1 AND 3
      P=P-(2*(C(I,ID)-C((I-1),ID)))+2*H
      ENDIF
      IF(IT.EQ.4)THEN
C     WETTED PERIMETER METHOD4
      P=P+(2*((C(I,ID)-C((I-1),ID))**2+W(K)**2)**0.5)
      ENDIF
      GOTO 98
C     LEFT HAND FLOODPLAIN
90    L=K+1
      GOTO 95
C     RIGHT HAND FLOODPLAIN
91    L=K-1
95    IF(IT.LT.3)GOTO 96
      AA=AA+((C(I,ID)-C((I-1),ID))*W(L)/2)
96    IF(IT.EQ.2)THEN
      P=P+(C(I,ID)-C((I-1),ID))
      ENDIF
98    R=AA/P
      IF(SEGN(K).LT.0.0)THEN
      SGN=-SEGN(K)
      GOTO 130
      ENDIF
      SGN=SEGN(K)
130   IF(MR.LT.1) GOTO 37
C     COMPUTE SEPARATE R.CURVES FOR EACH SEGMENT
      II=10*ID+K
      Q(I,II)=Q(I,II)+AA*R**.6667*SLPN/SGN
      A(I,II)=A(I,II)+AA
      GOTO 21
C     ADD DISCHARGES AND AREAS FOR ALL SEGMENTS TO OBTAIN TOTALS FOR
C     VALLEY SECTION.
37    Q(I,ID)=Q(I,ID)+AA*R**.66667*SLPN/SGN
      A(I,ID)=A(I,ID)+AA
21    CONTINUE
      JJJ=J-3
22    CONTINUE
      IF(ICODE.EQ.0)GO TO 29
      IF(MR.LT.1)GOTO 40
C     FIND MIN ELEV IN EACH SEGMENT
      J=13+2*NSEG
      DO 80 M=1,NSEG
      IF(M.EQ.1)THEN
      XM(10*ID+M)=DATA(ISG(M))
```

```
      GOTO 81
      ENDIF
      XM(10*ID+M)=DATA(ISG(M-1))
81    IF(J.GT.ISG(M))GOTO 80
      IF(DATA(J).LT.XM(10*ID+M))THEN
      XM(10*ID+M)=DATA(J)
      ENDIF
      J=J+2
      GOTO 81
80    CONTINUE
      DO 41 J=10*ID+1,10*ID+NSEG
      WRITE(6,47)J
      DO 42 I=1,20
      C1=C(I,ID)*0.3048
      A1=A(I,J)*0.093
      Q1=Q(I,J)*0.02832
      DEEP(I,J)=C(I,ID)-XM(J)
      IF(DEEP(I,J).LT.0)THEN
      DEEP(I,J)=0
      ENDIF
      WRITE(6,32) C1,A1,Q1
42    CONTINUE
41    CONTINUE
      GOTO 49
40    WRITE(6,31)VS
      DO 30 I=1,20
      C1=C(I,ID)*0.3048
      A1=A(I,ID)*0.093
      Q1=Q(I,ID)*0.02832
      DEEP(I,ID)=C(I,ID)-ELO
      WRITE(6,32)C1,A1,Q1
30    CONTINUE
      RETURN
29    IF(MR.LT.1)GOTO 43
C     FIND MIN ELEV IN SEGMENT
      J=13+2*NSEG
      DO 82 M=1,NSEG
      IF(M.EQ.1)THEN
      XM(10*ID+M)=DATA(ISG(M))
      GOTO 83
      ENDIF
      XM(10*ID+M)=DATA(ISG(M-1))
83    IF(J.GT.ISG(M))GOTO 82
      IF(DATA(J).LT.XM(10*ID+M))THEN
      XM(10*ID+M)=DATA(J)
      ENDIF
      J=J+2
      GOTO 83
82    CONTINUE
      DO 44 J=10*ID+1,10*ID+NSEG
      WRITE(6,48) J
      DO 45 I=1,20
      DEEP(I,J)=C(I,ID)-XM(J)
      IF(DEEP(I,J).LT.0)THEN
      DEEP(I,J)=0
      ENDIF
      WRITE(6,25) C(I,ID),A(I,J),Q(I,J)
      WRITE(6,*)
```

```
45    CONTINUE
44    CONTINUE
      GOTO 49
43    WRITE(6,24)VS
      DO 46 I=1,20
      DEEP(I,ID)=C(I,ID)-ELO
      WRITE (6,25) C(I,ID),A(I,ID),Q(I,ID)
      WRITE (6,*)
46    CONTINUE
      RETURN
C     COMPUTE % FLOW IN EACH SEGMENT
49    DO 50 I=10*ID+1,10*ID+NSEG
      DO 50 J=1,20
      TQ(J,ID)=TQ(J,ID)+Q(J,I)
50    CONTINUE
      DO 57 I=1,NSEG
      II=10*ID+I
      WRITE(6,59)II
      WRITE(6,*)
      DO 57 J=1,20
      PERQ(J,II)=Q(J,II)/TQ(J,ID)
      IF(J.EQ.1) THEN
      PERQ(J,II)=0
      ENDIF
      IF(PERQ(2,II).EQ.1.0)THEN
      PERQ(1,II)=1.0
      ENDIF
      IF(ICODE.GT.0)THEN
      C(J,ID)=C(J,ID)*0.3048
      ENDIF
      WRITE(6,60) C(J,ID),PERQ(J,II)
      WRITE(6,*)
      IF(ICODE.GT.0)THEN
      C(J,ID)=C(J,ID)/0.3048
      ENDIF
57    CONTINUE
      RETURN

24     FORMAT(1H0,T42,'RATING CURVE VALLEY SECTION ',F5.1/T46,'WATER',T56,
      &'FLOW',T66,'FLOW'/T45,'SURFACE',T56,'AREA',T66,'RATE'/T46,'ELEV',
      &T56,'SQ FT',T66,'CFS')
48     FORMAT(1H0,T42,'RATING CURVE FOR SEGMENT ',I5.1/T46,'WATER',T56,
      &'FLOW',T66,'FLOW'/T45,'SURFACE',T56,'AREA',T66,'RATE'/T46,'ELEV',
      xT56,'SQ FT',T66,'CFS')
31     FORMAT(1H0,T42,'RATING CURVE VALLEY SECTION',F5.1/T46,
      &'WATER',T56,'FLOW',T66,'FLOW'/T45,'SURFACE',T56,'AREA',
      &T66,'RATE'/T46,'ELEV',T56,'SQ M',T66,'CMS')
47     FORMAT(1H0,T42,'RATING CURVE FOR SEGMENT  ',I5.1/T46,
      &'WATER',T56,'FLOW',T66,'FLOW'/T45,'SURFACE',T56,'AREA',
      &T66,'RATE'/T46,'ELEV',T56,'SQ M',T66,'CMS')
25     FORMAT (40X,F10.2,2F10.1)
32     FORMAT (40X,3F10.2)
59     FORMAT (1H0,T47,'% DISCHARGE IN SEGMENT',I2.1/T46,
      &'ELEV',T55,'PERCENT')
60     FORMAT(40X,F10.2,2F10.3)
70     FORMAT(1H0,T10,'ERROR - NEED FLD PLAIN SEG BOTH S
      &IDES OF CHANNEL')
170    FORMAT(1H0,T20,'TURBULENT EXCHANGE METHOD',2X,I5.1)
```

```
      END


      SUBROUTINE STT

C     THIS SUBROUTINE STORES A DEPTH - FLOW - TRAVEL TIME TABLE.

      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20,6),
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NHD,NER,MAXNO,NCOMM,ICC,NCODE,TIME,KCODE,ICODE
      COMMON/BLOCK2/ PERQ(20,70),TQ(20,6),CC(20),LL(6),INRC,LRC

      ID=DATA(1)
      REACH=DATA(2)
      MET1=DATA(5)
      IF(MET1.EQ.0)GO TO 2
      DATA(3)=DATA(3)/0.3048
      J=6
      DO 3 I=1,19
      DATA(J)=DATA(J)/0.3048
      DATA(J+1)=DATA(J+1)/0.02832
3     J=J+3
2     XL=DATA(3)
      SLOPE=DATA(4)
      DIST(ID)=SLOPE*XL
      J=6
      DO 1 I=1,19
      DP(I)=DATA(J)
      SCFS(I)=DATA(J+1)
      CC(I)=DATA(J+2)
1     J=J+3
      RETURN
      END


      SUBROUTINE CMPTT

C     THIS SUBROUTINE COMPUTES THE TRAVEL TIME AT GIVEN
C     DISCHARGE RATES
C
C     IF MULTIPLE ROUTING INVOKED, COMPUTES TRAVEL TIME TABLE FOR
C     THE ONE SEGMENT SPECIFIED - OTHERWISE ALL SEGMENTS TOGETHER
C
C     NOTE -- FOR MULTIPLE ROUTINE NEED TO REPEAT THIS ROUTINE AND ROUTE
C     FOR "EACH" SEGMENT

      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20,6),
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NHD,NER,MAXNO,NCOMM,ICC NCODE,TIME,KCODE,ICODE
      COMMON/BLOCK2/ PERQ(20,70),TQ(20,6),CC(20),LL(6),INRC,LRC

      ID=DATA(1)
```

```
      REACH=DATA(2)
      NOVS=DATA(3)
      IF(KCODE.NE.0)DATA(4)=DATA(4)/0.3048
      XL=DATA(4)
      SLOPE=DATA(5)
      DIST(ID)=SLOPE*XL
      XLD36 = XL / 3600.
      MR=DATA(6)
C     MULTIPLE ROUTING
      INRC=DATA(7)
C     RATING CURVE AT TOP OF REACH
      LRC=DATA(8)
C     RATING CURVE AT DOWNSTREAM END
C     ZERO ARRAYS
      DO 1 J=1,20
      DATA (J)=0.
1     CFS(J)=0.
C     MULTIPLE ROUTING COMPUTATION
      IF(MR.LT.1)GOTO 30
      WRITE(6,37)
      ID1=INRC
      GOTO 2
30    ID1=1
C     FIND RATING CURVE WITH SMALLEST MAXIMUM FLOW RATE
2     QMIN=Q(20,ID1)
      MIN=ID1
      GO TO 4
31    ID1=LRC
      GOTO 32
3     ID1=ID1+1
32    IF (QMIN-Q(20,ID1)) 4,4,2
4     IF(MR.LT.1)GOTO33
      IF(ID1.EQ.INRC)GOTO 31
      IF(ID1.EQ.LRC)GOTO5
      WRITE(6,*)'ERROR only two r.curves allowed for m.routing'
      RETURN
33    IF (ID1-NOVS) 3,5,5
5     I=1
      LL(ID)=0
C     SET SCFS ARRAY EQUAL TO Q ARRAY OF LOWEST RATING CURVE
      DO 6 J=2,20
      SCFS(I)=Q(J,MIN)
      IF(MR.LT.1)GOTO 6
      IF(PERQ(J,MIN).LT.0.001)THEN
      LL(ID)=LL(ID)+1
      ENDIF
6     I=I+1
C     COMPUT END AREA AND DEPTH
      DO 9 ID1=1,NOVS
      IF(MR.LT.1) GOTO 34
      IF(ID1.EQ.1)THEN
      ID1=INRC
      GOTO 34
      ENDIF
      ID1=LRC
34    M=1+LL(ID)
      N=2+LL(ID)
      DO 36 J=M,19
```

```
      DO 7 I=N,20
      IF (Q(I,ID1)-SCFS(J)) 7,17,8
7     CONTINUE
17    DATA (J)=A(I,ID1)+DATA(J)
      CFS(J)=DEEP(I,ID1)+CFS(J)
      GO TO 36
8     XY=(SCFS(J)-Q(I-1,ID1))/(Q(I,ID1)-Q(I-1,ID1))
      DATA (J)=A(I-1,ID1)+XY*(A(I,ID1)-A(I-1,ID1))+DATA(J)
      CFS(J)=DEEP(I-1,ID1)+XY*(DEEP(I,ID1)-DEEP(I-1,ID1))+CFS(J)
36    CONTINUE
      IF(MR.LT.1) GOTO 9
      IF(ID1.EQ.LRC)GOTO 35
      ID1=1
9     CONTINUE
35    XNOVS=NOVS
      IF(ICODE.EQ.0)GO TO 19
      WRITE(6,20)REACH
      GO TO 21
19    WRITE(6,13)REACH
      ID1=MIN
21    DO 10 I=M,19
      AVAREA = DATA (I) / XNOVS
      DP (I) = CFS(I) / XNOVS
      S = AVAREA * XLD36
      CC(I)=S/SCFS(I)
      IF(SCFS(I).EQ.0) THEN
      CC(I)=0
      ENDIF
      IF(ICODE.EQ.0)GO TO 24
      DP1=DP(I)*0.3048
      SCFS1=SCFS(I)*0.02832
      WRITE(6,14)DP1,SCFS1,CC(I)
      GO TO 10
24    WRITE(6,14)DP(I),SCFS(I),CC(I)
10    CONTINUE
C     PUNCH CODE
      IF(NPU)12,12,25
25    IF(ICODE.EQ.0)GO TO 11
      XL1=XL*0.3048
      WRITE(7,22)ID,REACH,XL1,SLOPE,ICODE
      DO 23 I=1,19
      DP1=DP(I)*0.3048
      SCFS1=SCFS(I)*0.02832
      WRITE(7,26)DP1,SCFS1,CC(I)
23    CONTINUE
      RETURN
11    WRITE(7,15)ID,REACH,XL,SLOPE,ICODE
      WRITE (7,16) (DP(I),SCFS(I),CC(I),I=1,19)
12    RETURN
C
13    FORMAT(1H0,T46,'TRAVEL TIME TABLE'/T54,'REACH',F5.1//T46,'WATER',T
     &56,'FLOW',T65,'TRAVEL'/T46,'DEPTH',T56,'RATE',T66,'TIME'/T46,'FEET
     &',T56,'CFS',T66,'HRS')
14    FORMAT (40X,F10.2,F10.0,F10.2)
15    FORMAT('STORE TRAVEL TIME',T21,'ID=',I1,T29,'REACH NO=',F5.1,T44,
     &'LENGTH=',F9.0,' FT'/T21,'SLOPE=',F8.6,'FT/FT','CODE=',I1/T2
     &1,'DEPTH(FT)',T35,'FLOW(CFS)',T49,'TIME(HRS)')
20    FORMAT(1H0,T46,'TRAVEL TIME TABLE'/T54,'REACH',F5.1//T46,'WATER',T
```

```
     &56,'FLOW',T65,'TRAVEL'/T46,'DEPTH',T56,'RATE',T66,'TIME'/T46,
     &"METER",T56,'CMS',T66,'HRS')
22    FORMAT('STORE TRAVEL TIME',T21,'ID=',I1,T29,'REACH NO=',F5.1,T44,
     &'LENGTH=',F9.0,' M'/T21,'SLOPE=',F8.6,'M/M',"CODE=",I1/T2
     &1,'DEPTH(M)',T35,'FLOW(CMS)',T49,'TIME(HRS)')
16    FORMAT (T21,F7.2,F15.2,F15.3)
26    FORMAT (T21,F7.2,2F15.3)
37    FORMAT(1H0,T24,'MULTIPLE ROUTING INVOKED')
      END


      SUBROUTINE ROUTE

C     THIS SUBROUTINE ROUTES A HYDROGRAPH THROUGH A REACH WITH THE
C     NEW VSC METHOD OF FLOOD ROUTING  THIS METHOD ACCOUNTS FOR THE
C     VARIATION IN WATER SURFACE SLOPE.
C
C     IF MULTIPLE ROUTING INVOKED - COMPUTES PROPORTION INFLOW
C     FOR ONE SEGMENT
C
C     BUT  -----  ONLY ROUTES ONES SEGMENT AT A TIME
C     REPEAT TRAVEL TIME TABLE AND ROUTE COMMANDS FOR EACH SEGMENT
C     AND ADD OUTFLOWS

      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20,6),
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NHD,NER,MAXNO,NCOMM,ICC,NCODE,TIME,KCODE,ICODE
      COMMON/BLOCK2/ PERQ(20,70),TQ(20,6),CC(20),LL(6),INRC,LRC
      DIMENSION DOCFS(300,6)

C     New variables used
C     DOCFS copy inflow array (preserve original OCFS)
C     P percentage of OCFS (multiple routing)
C     DTDT copy inflow time increment (preserves DT(IDH))
C     TM increments DT
      ID=DATA(1)
      NHD=DATA(2)
      IDH=DATA(3)
      DT(ID)=DATA(4)
      DTDT=DT(IDH)
      DA(ID)=DA(IDH)
      M=IEND(IDH)
      MR=DATA(5)
C     CHECK: IF M.ROUTING ID.NE.IDH
      IF(MR.GT.0.AND.ID.EQ.IDH)THEN
      WRITE(6,*)'ERROR - FOR M.ROUTING ID MUST NOT BE SAME AS IDH'
      RETURN
      ENDIF
C     MULTIPLE ROUTING INCLUDED
C     SET UP DUMMY ARRAY
      DO 51 I=1,IEND(IDH)
51    DOCFS(I,IDH)=OCFS(I,IDH)
C     MULTIPLE ROUTING
C     COMPUTE DISTRIBUTED FLOW IN SEGMENT
      IF(MR.LT.1)GOTO 50
```

```
      II=INRC/10
      NN=INRC-10*II
      IF(ICODE.GT.0)GOTO 53
      WRITE(6,60)NN
      GOTO 55
53    WRITE(6,61)NN
55    TM=TIME-DTDT
      JJJJ=0
      DO 52 J=1,IEND(IDH)
      TM=TM+DTDT
      DO 56 K=2,20
      IF(DOCFS(J,IDH)-TQ(K,II))57,58,56
56    CONTINUE
      WRITE(6,*)'FAILED - RATING CURVE EXCEEDED'
      RETURN
58    DOCFS(J,IDH)=PERQ(J,INRC)*DOCFS(J,IDH)
      GOTO 54
57    ST=C(K,II)-(((TQ(K,II)-DOCFS(J,IDH))*(C(K,II)-C((K-1),II)))/(TQ
     &(K,II)-TQ(K-1,II)))
      P=PERQ(K,INRC)-(((C(K,II)-ST)*(PERQ(K,INRC)-PERQ((K-1),
     &INRC)))/(C(K,II)-C((K-1),II)))
      DOCFS(J,IDH)=P*DOCFS(J,IDH)
54    IF(DOCFS(J,IDH).EQ.0)THEN
      JJJJ=JJJJ+1
      IF(JJJJ.EQ.IEND(IDH))THEN
      WRITE(6,*)'NO FLOW IN SEGMENT'
      RETURN
      ENDIF
      GOTO 52
      ENDIF
      IF(ICODE.GT.0)THEN
      DOCFS(J,IDH)=DOCFS(J,IDH)*0.0283168
      ENDIF
      WRITE(6,59) TM,P,DOCFS(J,IDH)
      WRITE(6,*)
      IF(ICODE.GT.0)THEN
      DOCFS(J,IDH)=DOCFS(J,IDH)/0.0283168
      ENDIF
52    CONTINUE
C     IF ID AND IDH ARE EQUAL, ADD 1 TO IDH
50    IF(MR.LT.1)THEN
      LL(ID)=0
      ENDIF
      IF (ID-IDH) 3,1,3
1     IDH=IDH+1
      DO 2 I=1,M
2     DOCFS(I,IDH)=DOCFS(I,IDH-1)
      DT(IDH)=DT(IDH-1)
      PEAK(IDH)=PEAK(IDH-1)
3     NERRT=0
      PEAK(ID) = 1.
      RO = 0.
      N=19
      OCFS(1,ID)=0.
      S = 0.
      T1 = CC(1)
      J=1
      GUES = 1.
```

```
      CFS(1)=0.
C     IF ROUTING INTERVAL IS NOT EQUAL TO TIME INCREMENT OF INFLOW
C     HYDROGRAPH, INTERPOLATE
      IF (DT(ID)-DT(IDH)) 8,15,4
4     TID=DT(ID)
      TIDH=0.
      DO 7 I=2,M
      TIDH=TIDH+DT(IDH)
      IF (TID-TIDH) 6,5,7
5     J=J+1
      CFS(J)=DOCFS(I,IDH)
      TID=TID+DT(ID)
      GO TO 7
6     J=J+1
      CFS(J)=DOCFS(I-1,IDH)+((TID-TIDH+DT(IDH))/DT(IDH))*(DOC
     &FS(I,IDH)-DOCFS(I-1,IDH))
      TID=TID+DT(ID)
7     CONTINUE
      GO TO 13
8     TIDH=0.
      TID=DT(ID)
      DO 12 I=2,M
      TIDH=TIDH+DT(IDH)
9     IF (TIDH-TID) 12,10,11
10    J=J+1
      CFS(J)=DOCFS(I,IDH)
      TID=TID+DT(ID)
      IF (J-300) 12,13,13
11    J=J+1
      CFS(J)=DOCFS(I-1,IDH)+((TID-TIDH+DT(IDH))/DT(IDH))*(DOC
     &FS(I,IDH)-DOCFS(I-1,IDH))
      TID=TID+DT(ID)
      IF (J-300) 9,13,13
12    CONTINUE
13    DT(IDH)=DT(ID)
      M=J
      DO 14 I=2,M
14    DOCFS(I,IDH)=CFS(I)
C     IF INFLOW IS ZERO, SO IS OUTFLOW
15    DO 16 L=2,M
      IF (DOCFS(L,IDH)) 16,16,49
16    OCFS(L,ID)=0.
C     ROUTE
49    DATA (L-1) = 0.
      DO 42 I=L,300
      IF (I-M) 18,18,17
17    DOCFS(I,IDH)=DOCFS(I-1,IDH)*.9
18    AVIN=(DOCFS(I,IDH)+DOCFS(I-1,IDH))/2.
      SIA = AVIN + S
      J=1
C     DETERMINE DEPTH AND TRAVEL TIME OF INFLOW
      IF (DOCFS(I,IDH)-SCFS(1+LL(ID))) 19,23,20
19    DI2 = (DOCFS(I,IDH) / SCFS(1+LL(ID))) * DP(1+LL(ID))
      TI2 = CC(1+LL(ID))
      GO TO 25
20    JJJ=2
      IF(LL(ID).GT.0)THEN
      JJJ=LL(ID)+2
```

```
      ENDIF
      DO 21 J=JJJ,N
      IF (DOCFS(I,IDH)-SCFS(J)) 24,23,21
21    CONTINUE
      IF (NERRT) 22,22,36
22    WRITE (6,46)
      NERRT=1
      GO TO 36
23    DI2=DP(J)
      TI2 = CC(J)
      GO TO 25
24    RATIO=(DOCFS(I,IDH)-SCFS(J-1))/(SCFS(J)-SCFS(J-1))
      DI2=DP(J-1)+RATIO*(DP(J)-DP(J-1))
      TI2=CC(J-1)+RATIO*(CC(J)-CC(J-1))
25    DO 35 IT=1,10
      J=1
C     DETERMINE DEPTH AND TRAVEL TIME OF OUTFLOW
      IF (GUES-SCFS(1+LL(ID))) 26,29,27
26    DO2 = (GUES / SCFS(1+LL(ID)))* DP(1+LL(ID))
      TO2 = CC(1+LL(ID))
      GO TO 31
27    DO 28 J=JJJ,N
      IF (GUES-SCFS(J)) 30,29,28
28    CONTINUE
      J=N
29    DO2=DP(J)
      TO2=CC(J)
      GO TO 31
30    RATIO=(GUES-SCFS(J-1))/(SCFS(J)-SCFS(J-1))
      DO2=DP(J-1)+RATIO*(DP(J)-DP(J-1))
      TO2=CC(J-1)+RATIO*(CC(J)-CC(J-1))
C     FIND WATER SURFACE SLOPE
31    DDD=DIST(ID)/(DIST(ID)+DI2-DO2)
      IF (DDD-.01) 32,32,33
32    GUES=DOCFS(I-1,IDH)
      GO TO 35
33    T2 = .5 * (TI2 + TO2)
      T2=T2*SQRT(DDD)
      T = T1 + T2
C     COMPUTE ROUTING COEFFICIENT
      COEF =(2. * DT(ID)) / (T+DT(ID))
      O2 = COEF * SIA
      TRY1 = GUES
      RATIO=O2/(GUES+.1E-20)
      DIFF=ABS(1.-RATIO)
C     TEST FOR CONVERGENCE
      IF (DIFF-0.001) 37,37,34
34    GUES=O2
35    CONTINUE
      OCFS(I,ID)=DATA(I-1)*SIA
      DATA(I) = DATA(I-1)
      WRITE (6,47) I,OCFS(I,ID)
      GO TO 38
36    OCFS(I,ID)=DATA(I-1)*SIA
      DATA(I) = DATA(I-1)
      GO TO 38
37    OCFS(I,ID)=O2
      DATA (I) = COEF
```

```
C     COMPUTE NEW STORAGE
38    S = SIA - OCFS(I,ID)
      T1 = T2
      RO = RO + OCFS (I,ID)
      IF (OCFS(I,ID) - OCFS(I-1,ID)) 39,40,40
39    IF(OCFS(I,ID) -1.) 43,43,42
40    IF(OCFS(I,ID) - PEAK(ID)) 42,42,41
41    PEAK(ID)=OCFS (I,ID)
42    CONTINUE
      I=300
43    IEND(ID)=I
      ROIN(ID) = RO*DT(ID)*3600
C     PUNCH CODE
      IF (NPU) 45,45,44
44    WRITE (7,48) ID,NHD,IDH,DT(ID)
45    DT(IDH)=DTDT
      RETURN
C
46    FORMAT(1H0, 'TRAVEL TIME TABLE EXCEEDED')
47    FORMAT(T10,'PROBLEM FAILED TO CONVERGE AFTER10 ITERATIONS. CONVERG
     &ENCE WAS FORCED.'/T20,'OUTFLOW NUMBER = ',I4,'RATE =',F10.2)
48    FORMAT(   'ROUTE',T21,'ID=',I1,T29,'HYD NO=',I3,T45,'INFLOW ID=',I
     &1,T65,'DT=',F8.6,'HRS')
60    FORMAT(1H0,T40,'INFLOW FOR SEGMENT',I5.1/T30,'HOURS',T40,
     &'PERCENT',T52,'CFS')
61    FORMAT(1H0,T40,'INFLOW FOR SEGMENT',I5.1/T30,'HOURS',
     &T40,'PERCENT',T52,'CUMECS')
59    FORMAT(25X,F10.3,2F10.3,3F10.3)
      END


      SUBROUTINE RESVO

C     THIS SUBROUTINE ROUTES A HYDROGRAPH THROUGH A RESERVOIR WITH THE
C     STORAGE-INDICATION METHOD.

      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20,6),
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NHD,NER,MAXNO,NCOMM,ICC,NCODE,TIME,KCODE,ICODE
      COMMON/BLOCK2/ PERQ(20,70),TQ(20,6),CC(20),LL(6),INRC,LRC

      ID=DATA(1)
      NHD=DATA(2)
      IDH=DATA(3)
      NERES=0
      DT(ID)=DT(IDH)
      RO = 0.
      DA(ID)=DA(IDH)
      PEAK(ID) = 1.
      J=1
      I=4
C     REMAINING DATA ARE FLOW AND STORAGE VALUES
      IF(KCODE.EQ.0)GO TO 25
      DATA(I)=DATA(I)/0.02832
      DATA(I+1)=DATA(I+1)/1.21968
```

```
25    SCFS(J)=DATA(I)
      STORE1=DATA(I+1)*12.1
      STORE=STORE1
C     COMPUTE STORAGE COEFFICIENT ARRAY C
1     CC(J)=(SCFS(J)/2.)+(STORE/DT(ID))
      I=I+2
      J=J+1
      IF (J-20) 2,2,3
2     IF(KCODE.EQ.0)GO TO 26
      DATA(I)=DATA(I)/0.02832
      DATA(I+1)=DATA(I+1)/1.21968
26    SCFS(J)=DATA(I)
      STORE=DATA(I+1)*12.1
      IF (SCFS(J)-.001) 3,3,1
3     N=J-1
      OCFS(1,ID)=0.
      S=STORE1/DT(ID)
C     ROUTE
      DO 15 I=2,150
      IF (I-IEND(IDH)) 5,5,4
4     OCFS(I,IDH)=0.0
5     AVIN=(OCFS(I,IDH)+OCFS(I-1,IDH))/2.
      SIA=S+AVIN
C     DETERMINE PROPER C
      DO 6 J=1,N
      IF (SIA-CC(J)) 10,9,6
6     CONTINUE
      IF (NERES) 7,7,8
7     WRITE (6,19)
      NERES=1
8     RESC=SCFS(N)/CC(N)
C     COMPUT OUTFLOW
      OCFS(I,ID)=RESC*SIA
      GO TO 11
9     OCFS(I,ID)=SCFS(J)
      GO TO 11
10    OCFS(I,ID)=SCFS(J-1)+((SIA-CC(J-1))/(CC(J)-CC(J
     & -1)))*(SCFS(J)-SCFS(J-1))
C     DETERMINE NEW STORAGE
11    S=SIA-OCFS(I,ID)
      RO = RO + OCFS(I,ID)
      IF (OCFS(I,ID)-OCFS(I-1,ID)) 12,13,13
12    IF (OCFS(I,ID)-1.) 16,16,15
13    IF(OCFS(I,ID) - PEAK(ID)) 15,15,14
14    PEAK(ID) = OCFS(I,ID)
15    CONTINUE
      I=150
16    IEND(ID)=I
      ROIN(ID) = RO * DT(ID)*3600
C     PUNCH CODE
      IF (NPU) 18,18,17
17    II=2*N+3
      IF(ICODE.EQ.0)GO TO 22
      WRITE(7,24)ID,NHD,IDH,KCODE
      DO 23 I=5,II,2
      DATA(I)=DATA(I)*0.02832
      DATA(I+1)=DATA(I+1)*1.21968
23    CONTINUE
```

```
      WRITE(7,27)(DATA(I),I=5,II)
      RETURN
22    WRITE(7,20)ID,NHD,IDH,ICODE
      WRITE (7,21) (DATA(I),I=5,II)
18    RETURN
C
19    FORMAT (1H0,33HSTORAGE-DISCHARGE TABLE EXCEEDED.)
20    FORMAT(   'ROUTE RESERVOIR',T21,'ID=',I1,T29,'HYD NO=',I3,T42,'INF
     &LOW ID=',I1,T60,"CODE=",I1         /T21,'OUTFLOW(CFS)',T37,'STOR
     &AGE(AC FT)')
24    FORMAT(   'ROUTE RESERVOIR',T21,'ID=',I1,T29,'HYD NO=',I3,T42,'INF
     &LOW ID=',I1,T60,"CODE=",I1         /T21,'OUTFLOW(CMS)',T37,'STOR
     &AGE(1000CU M)')
21    FORMAT (T21,F10.1,F13.1)
27    FORMAT (T21,F10.2,F13.2)
      END


      SUBROUTINE ERROR

C This subroutine determines the error standard deviation and the peak flow
C error for 2 hydrographs (original program retained).
C Assumes that measured is ID1
C In addition, 10 other measures of goodness of fit are calculated.
C All indicies are printed out in metric units.

      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20,6),
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NHD,NER,MAXNO,NCOMM,ICC,NCODE,TIME,KCODE,ICODE
      ID1=DATA(1)
      ID2=DATA(2)
      SSE=0.
           WRITE(6,21)
21         FORMAT(1H0,T33,'TIME',T55,'FLOW 1',T76,
     &     'FLOW 2',T95,'ERROR'/T34,
     &     'HRS',T57,'CMS',T78,'CMS',T97,'CMS')

22    J=1
C If the time increments are not equal, interpolate.

      IF (DT(ID1)-DT(ID2)) 1,8,2
1     L=ID1
      K=ID2
      GO TO 3
2     L=ID2
      K=ID1
3     M=IEND(L)
      TID=DT(K)
      TIDH=0.
      DO 6 I=2,M
      TIDH=TIDH+DT(L)
      IF (TID-TIDH) 5,4,6
4     J=J+1
      CFS(J)=OCFS(I,L)
      TID=TID+DT(K)
```

```
      GO TO 6
5     J=J+1
      CFS(J)=OCFS(I-1,L)+((TID-TIDH+DT(L))/DT(L))*(OCFS(I,L)-OCFS(I-1,L)
     &)
      TID=TID+DT(K)
6     CONTINUE
      IEND(L)=J
      DT(L)=DT(K)
      DO 7 I=2,J
7     OCFS(I,L)=CFS(I)
8     IF (IEND(ID1)-IEND(ID2)) 9,9,10
9     M=IEND(ID1)
      GO TO 11
10    M=IEND(ID2)
11    T2=TIME

      IF (KCODE.EQ.0)THEN
        DO 997 I=1,M
              OCFS(I,ID1)=OCFS(I,ID1)*.02832
997           OCFS(I,ID2)=OCFS(I,ID2)*.02832
      ENDIF

C Determine error - original method

      DO 12 I=1,M
      ERR=OCFS(I,ID1)-OCFS(I,ID2)
              WRITE(6,16)T2,OCFS(I,ID1),OCFS(I,ID2),ERR
16            FORMAT (6X,F12.3,3F12.0)
25            T2=T2+DT(ID1)

C Sum of squares of error

12    SSE=SSE+ERR*ERR
      XM=M

C Error variance

      EVAR=SSE/XM

C Error standard deviation

      ESDEV=SQRT(EVAR)

C Percent error for peak discharge

      ERPK=ABS(PEAK(ID1)-PEAK(ID2))
      PCTER=(ERPK/PEAK(ID1))*100.

C Other goodness of fit calculations...

      SUM01=0.
      SUM0=0.
      SUM1=0.
      SUM2=0.
      SUM3=0.
      SUM4=0.
      SUM5=0.
      SUM6=0.
```

```
      SUM7=0.
      SUM8=0.
      SUM9=0.
      SUM10=0.
      SUM11=0.
      SUM12=0.

      DO 77 I=1,M
        ERR=OCFS(I,ID1)-OCFS(I,ID2)
        IF(OCFS(I,ID1).EQ.0.0.AND.OCFS(I,ID2).NE.0.0)THEN
            LOGERR=ALOG(OCFS(I,ID2))
        ELSE IF(OCFS(I,ID1).NE.0.0.AND.OCFS(I,ID2).EQ.0.0)THEN
            LOGERR=ALOG(OCFS(I,ID1))
        ELSE IF(OCFS(I,ID1).EQ.0.0.AND.OCFS(I,ID2).EQ.0.0)THEN
          LOGERR=0.
        ELSE
          LOGERR=ALOG(OCFS(I,ID1))-ALOG(OCFS(I,ID2))
        ENDIF
        SUM0=OCFS(I,ID1)+SUM0
        SUM01=OCFS(I,ID2)+SUM01
        SUM1=ERR+SUM1
        SUM2=ERR**2+SUM2
        SUM3=LOGERR**2+SUM3
        IF(OCFS(I,ID1).EQ.0.)OCFS(I,ID1)=1.
        SUM4=((ERR/OCFS(I,ID1))**2)+SUM4
77      CONTINUE

      DO 13 I=2,M
        DIFF1=OCFS(I,ID1)-OCFS(I-1,ID1)
        DIFF2=OCFS(I,ID2)-OCFS(I-1,ID2)
        SUM5=((DIFF1-DIFF2)**2)+SUM5
        SUM7=DIFF1+SUM7
        IF(DIFF1.EQ.0.)DIFF1=1.
        SUM6=(((DIFF1-DIFF2)/DIFF1)**2)+SUM6
13      CONTINUE


      SIMMEAN=SUM01/M
      OBSMEAN=SUM0/M
      DIFFM1=SUM7/M

      DO 14 I=2,M
        SUM8=(((OCFS(I,ID1)-OCFS(I-1,ID1))-DIFFM1)**2)+SUM8
        SUM9=((((OCFS(I,ID1)-OCFS(I-1,ID1))/DIFFM1)-1)**2)+SUM9
14      CONTINUE

      DO 73 I=1,M
        SUM10=((OCFS(I,ID1)-OBSMEAN)**2)+SUM10
        SUM11=(((OCFS(I,ID1)/OBSMEAN)-1)**2)+SUM11
        SUM12=((OCFS(I,ID2)-SIMMEAN)**2)+SUM12
73      CONTINUE

      SDM=SQRT(SUM10/(M-1))
      SDS=SQRT(SUM12/(M-1))

      DO 115 I=1,M
115       SUM13=((OCFS(I,ID1)-OBSMEAN)/SDM)*((OCFS(I,ID2)-
     &    SIMMEAN)/SDS)+SUM13
```

```
      OF1=SUM1
      OF2=SUM2
      OF3=SUM3
      OF4=SUM4
      OF5=SUM5
      OF6=SUM6
      OF7=SUM2/SUM10
      OF8=SUM4/SUM11
      OF9=SUM5/SUM8
      OF10=SUM6/SUM9
      AM=M
      OF11=(1./AM)*SUM13

      WRITE(6,95)
95    FORMAT(1H0,10X,'-----------------------------------------')
      WRITE(6,50)
50    FORMAT(15X,' MEASURES OF FIT '//)
      WRITE(6,91)
91    FORMAT(10X,'-----------------------------------------')
      WRITE(6,51)OF1
51    FORMAT(10X,'SUM OF ERRORS           ',F20.5)
      WRITE(6,52)OF2
52    FORMAT(10X,'OLSQ                    ',F20.5)
      WRITE(6,53)OF3
53    FORMAT(10X,'LOG LSQ                 ',F20.5)
      WRITE(6,54)OF4
54    FORMAT(10X,'RELATIVE ERROR          ',F20.5)
      WRITE(6,55)OF5
55    FORMAT(10X,'ABS ERROR - DIFF        ',F20.5)
      WRITE(6,56)OF6
56    FORMAT(10X,'REL ERROR - DIFF        ',F20.5)
      WRITE(6,57)OF7
57    FORMAT(10X,'ABS ERROR/VAR          ',F20.5)
      WRITE(6,58)OF8
58    FORMAT(10X,'REL ERROR/VAR          ',F20.5)
      WRITE(6,59)OF9
59    FORMAT(10X,'ABS ERROR(diff)/VAR     ',F20.5)
      WRITE(6,60)OF10
60    FORMAT(10X,'REL ERROR(diff)/VAR     ',F20.5)
      WRITE(6,61)OF11
61    FORMAT(10X,'PEARSONS r              ',F20.5)
      WRITE(6,92)ESDEV
92    FORMAT(10X,'ERR STANDARD DEV        ',F20.5)
      WRITE(6,93)PCTER
93    FORMAT(10X,'PEAK Q ERROR            ',F20.5)
      WRITE(6,96)
96    FORMAT(10X,'-----------------------------------------')

      WRITE (6,98)
98    FORMAT (//10X 'NOTE:  All indicies are in metric units')

      IF (KCODE.EQ.0)THEN
        DO 9969 I=1,M
              OCFS(I,ID1)=OCFS(I,ID1)/.02832
9969          OCFS(I,ID2)=OCFS(I,ID2)/.02832
      ENDIF
```

```
      RETURN
C
      END


      SUBROUTINE SEDT

C     THIS SUBROUTINE COMPUTES THE SEDIMENT YIELD FOR A FLOOD

      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20,6),
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NHD,NER,MAXNO,NCOMM,ICC,NCODE,TIME,KCODE,ICODE

      ID=DATA(1)
      SOIL=DATA(2)
      CROP=DATA(3)
      CP=DATA(4)
      SL=DATA(5)
      WRITE(6,*)'** CHECK THIS IS CORRECT AREA',DA(ID)
      WRITE(6,*)'ESPECIALLY IF MULTIPLE ROUTING UTILIZED'
C     COMPUTE SEDIMENT YIELD
      X=ROIN(ID)*DA(ID)*53.333*PEAK(ID)
      SED=95.*X**.56*SOIL*CROP*CP*SL
      IF(ICODE.EQ.0)GO TO 5
      SED1=SED*0.9072
      WRITE(6,6)SED1
      GO TO 7
5     WRITE (6,3) SED
C     PUNCH CODE
7     IF(NPU)2,2,1
1     WRITE (7,4) ID,SOIL,CROP,CP,SL
2     RETURN
3     FORMAT (10X, 'SEDIMENT YIELD = ', F10.1, ' TONS')
4     FORMAT(   'SEDIMENT YIELD',T21,'ID=',I1,T29,'SOIL=',F5.3,T42,'CROP
     &=',F5.3,T57,'CP=',F5.3,T70,'LS=',F5.3)
6     FORMAT(10X,"SEDIMENT YIELD=",F10.1,"METRIC TON")
      END


      BLOCK DATA

C     BLOCK DATA SUBPROGRAM UZED TO INITIALIZE ZALPHA,CTBLE,ITBLE
C     AND NCOMM.

      COMMON/BLOCK1/ OCFS(300,6),DATA(311),CFS(300),CTBLE(50,11),
     &RAIN(300),ROIN(6),
     &A(20,70),Q(20,70),DEEP(20,70),ITBLE(50,2),DP(20),SCFS(20),C(20,6),
     &ZALPHA(20),IEND(6),DA(6),DIST(6),SEGN(6),DT(6),PEAK(6),ISG(6),
     &NPU,NHD,NER,MAXNO,NCOMM,ICC,NCODE,TIME,KCODE,ICODE

      COMMON/BLOCK2/ PERQ(20,70),TQ(20,6),CC(20),LL(6),INRC,LRC

      DATA ZALPHA/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0,1H ,
```

```
&1H*,1H.,1H,,1H-,1H ,1H ,1H ,1H ,1H /

 DATA NCOMM/17/

 DATA CTBLE/1HS,1HS,1HR,1HC,1HP,1HP,1HP,1HA,1HS,1HC,1HS,1HC,1HR,
&1HR,1HE,1HS,1HF,33*1H ,
&1HT,1HT,1HE,1HO,1HR,1HU,1HL,1HD,1HT,1HO,1HT,1HO,1HO,1HO,1HR,
&1HE,1HI,33*1H ,
&2HAR,2HOR,2HCA,2HMP,2HIN,2HNC,2HOT,2HD ,2HOR,2HMP,2HOR,2HMP,
&2HUT,2HUT,2HRO,2HDI,2HNI,33*2H  ,
&2HT ,2HE ,2HLL,2HUT,2HT ,2HH ,2H H,2HHY,2HE ,2HUT,2HE ,2HUT,
&2HE ,2HE ,2HR ,2HME,2HSH,33*2H  ,
&2H  ,2HHY,2H H,2HE ,2HHY,2HHY,2HYD,2HD ,2HRA,2HE ,2HTR,2HE ,
&2H  ,2HRE,2HAN,2HNT,2H  ,33*2H  ,
&2H  ,2HD ,2HYD,2HHY,2HD ,2HD ,2H  ,2H  ,2HTI,2HRA,2HAV,2HTR,
&2H  ,2HSE,2HAL,2H Y,2H  ,33*2H  ,
&2H  ,2H  ,2H  ,2HD ,2H  ,2H  ,2H  ,2H  ,2HNG,2HTI,2HEL,2HAV,
&2H  ,2HRV,2HYS,2HIE,2H  ,33*2H  ,
&8*2H  ,2H C,2HNG,2H T,2HEL,2H  ,2HOI,2HIS,2HLD,34*2H  ,
&8*2H  ,2HUR,2H C,2HIM,2H T,2H  ,2HR ,36*2H  ,
&8*2H  ,2HVE,2HUR,2HE ,2HIM,38*2H  ,
&9*2H  ,2HVE,2H  ,2HE ,38*2H  /

 DATA ITBLE/1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,33*1H ,
&4,310,310,310,4,1,2,4,100,310,100,8,7,25,2,5,0,33*1H /
 END
```

```
* EXAMPLE data1 DATASET
START                 00.00 0 0 1
*  RUNOFF SUBCATCHMENT 406, MARBACH
*  SUBCATCHMENT 406
STORE HYD             ID=2  HYD NO=406  DT=2.0 HRS  DA=57.1 SQ MI
                      BASEFLOW = 154 CFS
                      FLOW RATES = 717 653 635 632 580 613
                      907 1095 950 483 295 322 296 173 99 65 63
                      67 43 27 18 13 10 7 5 4 3 2 85 185 98 42 18
                      13 9 20 33 18 9 103 274 239 185 315 356 186
                      141 168 99 53 31 23 101 197 107 147 238 131
                      61 30 22 16 12 9 7 5 4 3 2 1 1
* +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* ROUTE RUNOFF TO HERMANNSPIEGAL
* COMPUTE RATING CURVE FOR MARBACH
COMPUTE RATING CURVE ID=1  IT=1  MR=1 VS=6 NO SEGS=3
                      MIN ELEV=831.0FT  MAX ELEV=869.4FT
                      CH SLP=0.006  FLDPN SLP=0.0075
                      N=0.05 DIST=111.6FT
                      N=-.03 DIST=141.1FT
                      N=0.05 DIST=344.8FT
                       DIST     ELEV    DIST     ELEV
                        FT       FT      FT       FT
                        0.0     869.4   32.8     856.3
                       40.0     855.3   52.5     854.3
                       65.6     847.8   80.4     845.
                       91.9     841.9  105.0     837.9
                      111.6     836.9  114.8     835.3
                      121.4     831.0  134.5     831.0
                      141.1     836.6  144.4     836.9
                      157.5     839.6  170.6     840.2
                      180.4     845.5  344.8     859.6
*  COMPUTE RATING CURVE FOR HERMANNSPIEGAL
COMPUTE RATING CURVE  ID=2 IT=1 MR=1
                       VS=7 NO SEGS=3 MIN ELEV=687.7FT
                       MAX ELEV=702.1ft
                       CH SLP = 0.006  FLDPL SLP=0.0075
                       N=0.05  DIST=118.4FT
                       N=-.03  DIST=171.6FT
                       N=0.05  DIST=220.1FT
                       DIST    ELEV     DIST   ELEV
                        FT      FT       FT     FT
                         0.0   702.1     49.2  697.5
                       105.6   696.5    118.4  696.2
                       124.7   693.9    132.2  688.3
                       133.5   688.3    136.2  687.7
                       147.6   688.0    154.2  688.3
                        .6.5   689.0    157.5  689.6
                       167.3   695.5    171.6  696.2
                       187.0   696.2    200.1  696.5
                       212.3   696.2    216.5  698.5
                       220.1   722.1
* LEFT SEGMENT
COMPUTE TRAVEL TIME    ID=3  REACH NO=4  NO VS=2
                       L=110564.3 FT  SLP=0.007 MR=1
                       INRC=11  LRC=21
ROUTE                  ID=3  HYD NO=141  INFLOW ID=2
                       DT=0.5 HRS MR=1
```

```
* CHANNEL SEGMENT
COMPUTE TRAVEL TIME   ID=4  REACH NO=4   NO VS=2
                      L=110564.3 FT  SLP=0.007 MR=1
                      INRC=12 LRC=22
ROUTE                 ID=4  HYD NO=142  INFLOW=2
                      DT=0.5 HRS  MR=1
* RIGHT SEGMENT
COMPUTE TRAVEL TIME   ID=5  REACH NO=4 NO VS=2
                      L=110564.3 FT  SLP=0.007 MR=1
                      INRC=13 LRC=23
ROUTE                 ID=5  HYD NO=143  INFLOW=2
                      DT=0.5 HRS  MR=1
* ADD SEGMENTS
ADD HYD               ID=2  HYD NO=407  IDI=3 IDII=4
ADD HYD               ID=2  HYD NO=407  IDI=2 IDII=5
* ------------------------------------------------------------
*  RUNOFF SUBCATCHMENT 407
*  SUBCATCHMENT 407
STORE HYD             ID=3  HYD NO=407  DT=2.0 HRS  DA=105.7 SQ MI
                      BASEFLOW = 83 CFS
                      FLOW RATES= 1022 1155 1318 1403 1401 1320
                      1380 1698 2019 1807 1261 840 775 717 550 389
                      281 210 160 133 112 93 78 66 55 46 38 32 56
                      187 248 197 122 75 47 53 155 203 161 124 217
                      354 382 458 618 602 448 414 478 433 305 199
                      143 154 157 191 456 576 452 279 173 109 71 58
                      48 40 34 28 24 20 16 14 11 9 8 6 5 4 4 3 3 2 1
* ------------------------------------------------------------
*  OUTFLOW HERMANNSPIEGAL
ADD HYD               ID=2  HYD NO=407  IDI=3  IDII=2
PRINT HYD             ID=2  NPK=1 IDR=0 IN=0
FINISH
```